Quantization using jet space geometry and identity management using credential schemes

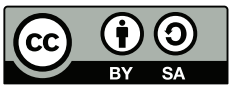
Ringers, Sietse

Quantization using Jet Space Geometry and Identity Management using Credential Schemes

Sietse Ringers

This PhD project was carried out at the Johann Bernoulli Institute of the University of Groningen, and the Institute for Computing and Information Sciences of the Radboud University. It was financially supported by the FWN / JBI RUG and the Secure Self-Enrollment (SSE) project from KPN.

Cover page: a Calabi-Yau manifold processed by a machine learning algorithm called "Deep Style"



ISBN: 978-90-367-9113-7 (printed version)
ISBN: 978-90-367-9112-0 (electronic version)

university of groningen

Quantization using Jet Space Geometry and Identity Management using Credential Schemes

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. E. Sterken
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Friday 7 October at 14:30 hours

by

Sietse Ringers

born on 11 July 1984
in Lelystad

Supervisor
Prof. J. Top

Co-supervisors
Dr. A. V. Kiselev
Dr. J.-H. Hoepman

Assessment Committee
Prof. G. R. Renardel de Lavalette
Prof. S. Gutt
Prof. B. de Decker

Contents (chapters)

Contents (detailed)

Introduction

Preface

This thesis consists of two parts: one that deals with geometrical aspects of differential equations and the mathematical side of two ways of quantizing physical theories, and one that deals with identity management and credential schemes in computer science. In the sections below, for each part we present an introduction that gives some motivation and historical context, and highlights some common themes between the chapters. Following that there are summaries that describe in more detail the contents of the two parts, and what we have contributed to their topics. (A Dutch version is included, starting on p. 187.)

Part 1: Quantization using jet space geometry

Introduction

If there is one thing that mathematicians and physicists (and also computer scientists) have in common, it is their appreciation and desire for aesthetics: both mathematicians and physicists strive for that perfect proof or the most elegant theory. Although they do not always agree on what is aesthetically pleasing and what is not, in overlapping areas such as mathematical physics their interests and tastes seem to align more often than not. This can lead to very interesting cross-fertilization.

A theme that occurs throughout the first part of this thesis, and which finds appreciation from people from both parties, is that of specialization: applying a general theory to a well-understood specific case, obtaining a special case of the theory. This special case of the more general theory is often well-known. Actually, it is often more interesting to think about going in the other direction: having some theory, one can try to go the other way and think about possible generalizations. This can be a guiding and inspiring

principle in the construction and understanding of new theories. Physically, traversing this path has led to ever more powerful theories that can describe and predict progressively more phenomena, while simultaneously the mathematical sides of the theories have grown more clear, powerful and mature as well. This leads to the following research question, which we will deal with throughout this part of the thesis: *How can the principles of generalization and specialization be used to deepen our understanding of the connections between mathematics and theoretical physics, in particular in the fields of jet space geometry and of quantization methods, and to what extent can this advance the mathematical side of these connections?*

Let us try to make this more tangible by considering two examples, of which the first corresponds to Chapters 1 and 2. Most of the partial differential equations occurring in physics, and many originating from mathematics as well, can be described elegantly in terms of infinite jet bundles. The independent variables together form a base space, while the unknowns form the fiber of a fiber bundle over the base space. The differential equation can then be described as a submanifold of the infinite jet bundle associated to the fiber bundle. Physically, such systems allow one to describe the dynamics of, for example, fields, waves or strings. When one applies this machinery to a point particle, however, the partial differential equation becomes an ordinary one, the base space collapses to a point and the infinite jet bundle to a (finite-dimensional) smooth manifold. The physics of point-particles is well-understood, and is formulated in terms of the geometry that comes naturally with manifolds, such as vector fields and the De Rham differential. Under the "specialization map" that sends the general theory to the case of the point particle, one can then wonder what the inverse of this geometrical machinery is. Essentially, the idea that there should be such inverses under this specialization map, and that they should play similar roles in the general physics theories as they do in the specialized ones, is the mathematical version of the idea from physics that the point-particle case should be a special case of the general framework. This idea is called secondary calculus, and although it plays an important role throughout the first part of this thesis it features most prominently in Chapter 2 (see Section 2.2 on p. 26.)

The second example is the topic of Chapter 3. In physics, quantum mechanics revolves around the Planck constant $\hbar \approx 1.055 \times 10^{-34}$ Js. Disregarding the constant nature of the Planck constant for the moment, quantum mechanical theories are generally such that when one takes all formulas and applies the limit $\hbar \to 0$, one obtains some classical approximating theory. Mathematically, it makes sense to think of the Planck constant as a deformation parameter, so that the quantum theory is a deformation of the classical theory. This is the essential idea behind quantization deformation. One starts with the classical theory of a point particle that lives on some manifold M, whose dynamics is described by a Poisson structure and a Hamiltonian vector field. Physically, the space of observables of this theory is the ring of functions $C^\infty(M)$. Then one interprets the quantum mechanical space of observables as a deformation in $\hbar$ of $C^\infty(M)$, leading to a non-commutative product called the star product, which contains not only the original pointwise product of functions on $C^\infty(M)$ but also the Poisson structure in terms of which the dynamics is described. This leads to an elegant quantum theory of point particles, which is such that if one takes the limit $\hbar \to 0$ one re-obtains the classical theory that was our starting point. (This method of quantizing classical physics theories

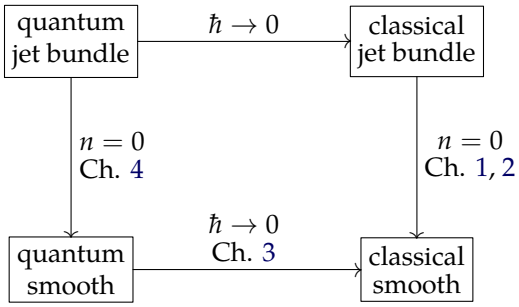


Figure 1. A schematic overview of the chapters and their topics. The text in the vertices indicates whether the theory is quantum mechanical or classical, and if it lives on smooth manifolds or jet bundles. An arrow indicates that one can specialize the theory to the one the arrow points at, by taking the limit prescribed by the label of the arrow (n refers to the dimension of the base space). Note that the chapters essentially travel in the opposite direction.

is certainly not the only one; in the physics community the BRST- and BV-formalisms seem to be more popular, as they are central to the highly successful standard model.)

Figure 1 shows the relations between the chapters. We see that both Chapters 2 and 3 start from a classical theory of a point particle living on a manifold, and then generalize this to other subjects than point particles (i.e., to jet bundles) and to a quantum theory (i.e., deformation quantization), respectively. This leads to a classical theory that can describe many physical phenomena on the one hand, and a quantum theory of point particles on the other hand. One can then wonder if both of these can in turn be considered as the specialization of some quantum theory of more general physical configurations than point particles – that is, of the theory that would be in the top-left corner of the diagram above. Mathematically, this would be a deformation quantization theory on jet bundles. We briefly consider this question in Chapter 4.

Abstract

The chapters of this first part of the thesis are organised as follows.

Chapter 1 gives a brief introduction to infinite jet bundles and a number of related geometrical constructions, since these occur in most of the first part of this thesis. After that we describe multivectors on jet bundles as our first application of secondary calculus. These constitute the domain of the variational Schouten bracket, which takes two such multivectors and returns a new one; we explain how this bracket is a natural extension of the Schouten bracket on smooth manifolds. The variational Schouten bracket has appeared in multiple guises in the mathematical and physical literature over the years, leading to a number of different descriptions of the same concept. In the final section of this chapter, we show how each of these different versions are actually equivalent, when care is taken. This chapter is based on the following article.

[KR12] A. V. Kiselev and S. Ringers. "A comparison of definitions for the Schouten bracket on jet spaces". In: *Proceedings of the Sixth International Workshop "Group Analysis of Differential Equations and Integrable Systems"*. Larnaca, Cyprus, 2012, 15p. arXiv: 1208.6196.

Chapter 2 is concerned with the Batalin-Vilkovisky formalism, which is one of the techniques for achieving the quantization of classical physics theories. Together with the variational Schouten bracket discussed in the first chapter, an important ingredient of this formalism is the BV-Laplacian: a differential operator of order two that should square to zero, and that together with the bracket forms a mathematical structure known as a BV-algebra. A problem, however, with the BV-Laplacian as it is usually defined within the physics literature, is that it contains "infinite constants" or delta-functions that then need to be removed by regularization. After having cast the geometrical setup in terms of the infinite jet bundles that were introduced in the previous chapter, we briefly explore the possibility of defining a BV-Laplacian that does not contain such infinite constants. We will see, however, that the obvious way of doing this does not result in a BV-algebra.

Chapter 3 revolves around deformation quantization, an entirely different method for the quantization of classical physics theories (but only for point particles). We first briefly explain the physical formalism of point particles in terms of Poisson structures and Hamiltonian vector fields on smooth manifolds. Using Kontsevich' celebrated result on the deformation quantization of Poisson manifold, any such theory can be converted into a quantum mechanical one, whose main ingredient is a noncommutative but associative product, called the star product. After having defined the notion of star products and the one that Kontsevich found for arbitrary Poisson manifolds, we give a thorough explanation of this formalism applied to the natural Poisson structure on the dual of Lie algebras.

Chapter 4 briefly discusses possible ways of combining secondary calculus in the form of infinite jet bundles with deformation quantization. Within the framework of secondary calculus, Poisson structures and Hamiltonian vector fields have generalizations to the setup of infinite jet bundles. One can then wonder if there is such a thing as a variational deformation quantization: a generalization of star products to the variational setup. Physically, such theories might be a generalization of the deformation quantization method from point particles to more general configurations. There are a number of possible generalizations that one can try, of which we show in this final chapter that none of these work.

Prerequisites

At the very least the reader should be familiar with smooth manifolds and the various structures and machinery surrounding them; some familiarity with symplectic and Poisson manifolds will help. There is a very brief introduction to infinite jet spaces in Chapter 1, but this serves more to fix notation than to truly explain these matters; it would be best if the reader already has some knowledge on these matters (see the section below for references). In principle no prior knowledge of physics is required; all

relevant physical concepts are introduced in the text (although, again, familiarity with these subjects will help).

Bibliographic notes

The rather miraculous observation that nature always seems to allow an accurate mathematical description – or conversely, that mathematics somehow always manages to capture physical theories so precisely, has been described beautifully by, among others, Wigner and Hamming in the following two articles.

[Wig60] E. P. Wigner. "The unreasonable effectiveness of mathematics in the natural sciences". In: *Comm. Pure Appl. Math.* 13 (Feb. 1960). URL: https://www.dartmouth.edu/~matc/MathDrama/reading/Wigner.html.

[Ham80] R. W. Hamming. "The unreasonable effectiveness of mathematics". In: *Amer. Math. Monthly* 87.2 (Feb. 1980). URL: https://www.dartmouth.edu/~matc/MathDrama/reading/Hamming.html.

For more on infinite jet bundles, their geometry, and how they relate to partial differential equations we refer to the following books and articles, the last of which introduces the concept of secondary calculus.

[KV99] I. S. Krasil'shchik and A. M. Vinogradov, eds. *Symmetries and conservation laws for differential equations of mathematical physics*. Vol. 182. Translations of Mathematical Monographs. Providence, RI: Amer. Math. Soc., 1999, pp. xiv+333.

[KV11] I. S. Krasil'shchik and A. Verbovetsky. "Geometry of jet spaces and integrable systems". In: *J. Geom. Phys.* 61.9 (2011), pp. 1633–1674. arXiv: 1002.0077.

[Kis12c] A. V. Kiselev. "The twelve lectures in the (non)commutative geometry of differential equations". 140p, preprint IHÉS M-12-13. 2012. URL: http://preprints.ihes.fr/2012/M/M-12-13.pdf.

[Olv00] P. J. Olver. *Applications of Lie groups to differential equations*. Second Edition. Vol. 107. Graduate Texts in Mathematics. New York: Springer-Verlag, 2000, p. 541.

[Vin01] A. M. Vinogradov. *Cohomological Analysis of Partial Differential Equations and Secondary Calculus*. Vol. 204. Translations of Mathematical Monographs. Amer. Math. Soc., 2001.

The BRST- and the BV-formalisms were introduced in the following papers.

[BRS75] C. Becchi, A. Rouet, and R. Stora. "Renormalization of the abelian Higgs-Kibble model". In: *Commun. Math. Phys.* 42 (June 1975), pp. 127–162.

[BRS76] C. Becchi, A. Rouet, and R. Stora. "Renormalization of gauge theories". In: *Ann. Phys.* 98 (June 1976), pp. 287–321.

[Tyu75] I. V. Tyutin. "Gauge Invariance in Field Theory and Statistical Physics in Operator Formalism". Lebedev Physics Institute preprint 39. 1975.

[BV81] I. A. Batalin and G. A. Vilkovisky. "Gauge algebra and quantization". In: *Phys. Lett. B* 102 (June 1981), pp. 27–31.

[BV83] I. A. Batalin and G. A. Vilkovisky. "Quantization of gauge theories with linearly dependent generators". In: *Phys. Rev. D* 28 (10 Nov. 1983), pp. 2567–2582.

The origin of the concept of deformation quantization seems to be [Bay+77; Bay+78a; Bay+78b]. The deformation quantization of smooth Poisson manifolds was achieved by Kontsevich in [Kon03]; before that Fedosov proved that all symplectic manifolds can be deformed quantized [Fed94].

[Bay+77] F. Bayen, M. Flato, C. Fronsdal, A. Lichnerowicz, and D. Sternheimer. "Quantum mechanics as a deformation of classical mechanics". In: *Letters in Mathematical Physics* 1.6 (1977), pp. 521–530.

[Bay+78a] F. Bayen, M. Flato, C. Fronsdal, A. Lichnerowicz, and D. Sternheimer. "Deformation theory and quantization. I. Deformations of symplectic structures". In: *Annals of Physics* 111.1 (1978), pp. 61–110.

[Bay+78b] F. Bayen, M. Flato, C. Fronsdal, A. Lichnerowicz, and D. Sternheimer. "Deformation theory and quantization. II. Physical applications". In: *Annals of Physics* 111.1 (1978), pp. 111–151.

[Kon03] M. Kontsevich. "Deformation Quantization of Poisson Manifolds". In: *Letters in Mathematical Physics* 66.3 (2003), pp. 157–216. arXiv: `q-alg/9709040`.

[Fed94] B. V. Fedosov. "A simple geometrical construction of deformation quantization". In: *J. Differential Geom.* 40.2 (1994), pp. 213–238.

Part 2: Identity management using credential schemes

Introduction

In recent years, the importance of information in our society has increased at staggering rates, primarily because of the arrival and omnipresence of the (personal) computer and the internet. More and more of the things we do happens online in one way or another. People use the internet to stay in touch with each other, to meet new people, to buy and sell, to schedule appointments, and to file their tax returns, to name a few. All of these actions generate information that can be stored and analyzed. In addition, since a few years nearly everyone has a smart phone on their person at all times, constantly connected to the internet, enabling firstly the constant and automatic recording of all sorts of data – for example, sound, location and movement, but also clicks or taps, purchases, or social relations – and secondly the easy and efficient transmission of this information through the internet. At the same time, the cost of transmitting and storing data is decreasing rapidly, and modern databases can easily store and quickly provide access to vast amounts of data.

Information about properties and activities of individuals, or *personal information* for short, is particularly important. In my case, this includes basic information like my age, name and where I live, but also that I have an interest in classical music, the fact that I purchased an e-book some weeks back, and my relationship status. It also includes past activities, as well as additional relevant information to these activities, such as spoken or written text in one form or another, financial records, or implications such as that I am interested in cryptography.

Considering the value of personal information along with the ease with which it can now be collected and stored, it is no surprise that many parties have started collecting it, in various ways and for various purposes. For example, the core business of Facebook and Google is to gather information about people to sell to advertisers; and a number of institutions of national security have turned out to tap from the internet as much data as they are able to take.

In some cases this monitoring of behavior and activities – i.e., *surveillance* – is what we want. For example, tax collection agencies necessarily need to be informed of people's incomes and savings, and many people routinely and purposefully share all sorts of data with the rest of the world on Facebook. In other cases there may be severe moral or legal issues, for example when a repressive government attempts to use surveillance to stifle dissent. People do not always have the option to do anything about it, even if they do not agree; for example, as someone who is not a citizen of the United States, I have no legal way to control or prevent the NSA from gathering information about me by tapping and analyzing internet traffic. People need not even be aware of it; the extent of the surveillance through the internet by agencies such as the NSA and GCHQ was largely unknown before the disclosures by Edward Snowden in 2013, and most people are largely unaware or do not think about how much of their personal information they give away when routinely interacting with the internet. It also often happens that an institution or company gathers much more data than it strictly needs in order to perform the services that it is expected to perform. It is clear, then, that surveillance can threaten people's privacy in many ways, and it is dangerous at least for the following three additional reasons.

- If whatever you do is always recorded by large companies and national institutions, these entities might be able to use this information against you in the future. Many of such entities claim that they can be trusted with all this information and responsibility, but even if this is true right now, there is no telling what they will be like in the future. Perhaps we live in a totalitarian state fifty years from now; if such a state knows everything about everyone there would be no escaping it.
- Once data about what you do or who you are is stored outside your control, it is very difficult to keep track of where and how it is stored, what is done with it, and who controls it. For example, privacy policies that state what is done with customer data are often subject to change without notice; the government could in the future implement (secret) laws or regulations that grants it access to such data; or the data may be stolen or leaked. Since such data can nowadays be stored essentially indefinitely, the risk of these things happening at some point in the future is significant, and indeed all three examples mentioned above have

happened a number of times by now. This issue also magnifies the risk described in the first point above.
- People may refrain from doing what they would otherwise do, knowing that they are watched online. For fear of potential future repercussions dissenting opinions and actions are thus repressed, leading to self-censorship. This is thoroughly unhealthy in an open and democratic society.

Summarizing, in a society where information equals money and power, it is unwise to heedlessly give your personal information away. It is a resource, and one that is constantly gaining importance at that. For these reasons, we believe it is becoming increasingly important that people are aware of and in control of who knows what about them. There is, in other words, a need for *identity management*: the process of obtaining, using and protecting data that deals with ones identity, online or offline, generally for the purpose of providing access to services. Note that we are not looking for ways to constantly hide all personal information, because this simply would not be possible: for example, a customer in a liquor store will somehow need to convince the show owner of the fact that she is over 18. However, there is no need for the shop owner to learn anything about her other than that fact in order to sell her the liquor.

Facts about persons such as that the customer is over 18, or that she has a subscription to some newspaper are called *attributes*. In fact, a passport can be seen as a list of such attributes created in an unforgeable way by the government. The example above illustrates that we want to be able to selectively disclose attributes, so that the customer in the liquor store can prove that she is indeed over 18, without revealing her precise age, name or other details that are not relevant to the purchase.

In the second part of this thesis, we will attempt to achieve these goals using a cryptographic technique called *attribute-based credential schemes*. In such a scheme a party called the issuer can grant credentials to users. Each credential can contain multiple attributes (which can store a limited amount of data), along with a signature created by the issuer over the attributes. The user can then selectively show some of his attributes, keeping the others hidden. The receiving party (often called the verifier) can use the signature to ensure that the credential was indeed created by the issuer. For example, using one such credential issued to me by the government, I could disclose all my personal information to a border guard at the airport, or just prove that I am over 18 to a shop owner.

Some credential schemes additionally offer an attractive feature called *multi-show unlinkability*: when two credentials are shown to a verifier, then the verifier cannot tell if he was shown one and the same credential twice, or two different ones (as long as the disclosed attributes, if any, do not tell the credentials apart). A credential scheme can separately or simultaneously be *issuer unlinkable*, preventing the credential issuer from linking credential usage sessions to specific credentials that it issued.

We will primarily focus on the following properties that credential schemes can offer.

Security. In order for the verifier to be truly convinced that the user owns a valid credential (more specifically, that this credential was given to the user by the issuer), it should be provable that users cannot create credentials on their own,

without the issuer's knowledge or consent. Similarly, whenever a scheme claims to offer certain anonymity features this too should be provable.

Efficiency. A credential scheme should be easy to use in practice. One way to achieve this could be to implement it on smart cards, which have highly limited resources. Therefore it makes sense to try to maximize the efficiency of credential schemes.

Attribute-based. As noted above, in different situations one might want to disclose different attributes. We will, however, also study some schemes that do not offer attributes, or indeed any storage capacity at all.

Anonymity. Whenever possible, a credential scheme should hide as much of the user as possible, or as much as the user wishes. Although this can already partly be achieved by hiding irrelevant attributes, we will mostly be interested in schemes that offer one and preferably both kinds of unlinkability.

Many of the schemes that we will study make use of bilinear group pairs: that is, a pair of two prime-order elliptic curves that admit a special map called the pairing. Elliptic curves offer high security levels combined with small group elements and fast algorithms for the group operations, while pairings give the ability to inspect relations between specific group elements. Thus the second part of this thesis deals with the following research question: *Does there exist or can we create a credential scheme in the setting of bilinear group pairs, that achieves all of the objectives stated above, including unlinkability? To what extent is there a trade-off between these objectives?*

The credential schemes that we study will achieve progressively more of the four goals above simultaneously. In Chapters 6 and 8 we study a number of existing schemes that were meant to be efficient, unlinkable and unforgeable, but failed. We will indeed find a trade-off between anonymity and efficiency, in the contrast between the two attribute-based credential schemes that we define and study in Chapters 7 and 9; the former offers high efficiency but no unlinkability, while the latter is not quite as efficient, but still finally achieves all four goals.

Abstract

The chapters of this second part of the thesis are organised as follows.

Chapter 5 introduces some basic notions and notations that we will use throughout this part of the thesis: what it means for something to be or not to be efficiently computable, cyclic groups and bilinear pairings, and zero-knowledge proofs and signature schemes. This chapter also introduces and defines credential schemes, the main topic of the subsequent chapters.

Chapter 6 studies a number of (non-attribute-based) credential schemes from the literature, which are broken in the sense that they do not offer the features they claim to offer. We prove a general theorem saying that in certain circumstances, it is very difficult to have such credential schemes provide both unlinkability and unforgeability. This chapter is based on the following article.

[HLR15] J.-H. Hoepman, W. Lueks, and S. Ringers. "On Linkability and Malleability in Self-blindable Credentials". In: *Information Security Theory and Practice: 9th IFIP WG 11.2 International Conference, WISTP 2015*. Ed. by N. R. Akram and S. Jajodia. Cham: Springer International Publishing, 2015, pp. 203–218.

Chapter 7 introduces an interactive signing algorithm for a generalization of the Boneh–Boyen signature scheme, that hides part of the message and the resulting signature from the signer. As an application, we describe a new attribute-based credential scheme, that is simpler and very efficient, at the cost of not being unlinkable (although the issuer is prevented from linking credentials that it issued with executions of the showing protocol). Compared with its most well-known competitor, the U-Prove credential scheme, this scheme offers similar efficiency but higher security guarantees.

Chapter 8 discusses the attribute-based scheme by Hanzlik and Kluczniak [HK14]. This scheme was meant to be an unlinkable version of U-Prove, which (like our scheme from Chapter 7) is not unlinkable but very efficient. Unfortunately, we have discovered an attack in Hanzlik and Kluczniak's scheme which allows colluding users to forge credentials over any set of attributes that they wish, without involvement or knowledge of the issuer. In this chapter we describe this attack, and we point out the error in the paper by Hanzlik and Kluczniak. This chapter is based on the following article.

[VRH16] E. Verheul, S. Ringers, and J.-H. Hoepman. "The self-blindable U-Prove scheme from FC'14 is forgeable". In: *Financial Cryptography and Data Security – FC'16* (2016). In print. URL: https://eprint.iacr.org/2015/725.

Chapter 9 finally introduces a second attribute-based credential scheme that does offer unlinkability, and which achieves all four goals stated in the previous section. Its security is guaranteed by our unforgeability proof. We also prove that the scheme is unlinkable, so that it protects the user's anonymity as much as possible. Finally, although the scheme is not as efficient as the one from Chapter 7, it is more efficient than any comparable (in particular, unlinkable) scheme that we know of. At the end of the chapter, we briefly compare our scheme with the ones from the previous chapters.

Bibliographic notes

The Data Protection Directive of the European Union limits the collection and usage of personal data for companies and institutions operating within the European Union.

[EUDPD] *Data Protection Directive*. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. See also http://ec.europa.eu/justice/data-protection/. URL: http://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:31995L0046.

Many of the activities of the NSA, GCHQ and a number of other such institutions were leaked by Edward Snowden in 2013, and subsequently published by news media such as *The Guardian*, *The New York Times*, *The Washington Post*, *Der Spiegel* and *The Intercept*. A complete archive of all these documents is maintained by the Canadian Journalists for Free Expression (CJFE).

[CJFE] *Snowden Surveillance Archive*. Canadian Journalists for Free Expression. URL: https://cjfe.org/snowden.

This thesis will mostly be concerned with the cryptographic side of identity management using credential schemes. There are many more aspects to this subject, ranging from what identity, privacy and anonymity really mean and are, to societal and legal implications. The second chapter of the PhD thesis of G. Alpár, as well as the paper that this chapter was based on, contain an extensive discussion about these matters.

[AHS11] G. Alpár, J. Hoepman, and J. Siljee. "The Identity Crisis. Security, Privacy and Usability Issues in Identity Management". In: *CoRR* abs/1101.0427 (2011). URL: http://arxiv.org/abs/1101.0427.

[Alp15] G. Alpár. "Attribute-Based Identity Management: Bridging the Cryptographic Design of ABCs with the Real World". PhD thesis. Radboud University, Nijmegen, The Netherlands, 2015.

The two best-known attribute-based credential schemes are U-Prove by S. Brands (and now owned by Microsoft), and Idemix by J. Camenisch and A. Lysyanskaya (and now owned by IBM). The former is very efficient while the latter offers more anonymity features and stronger security guarantees.

[Bra00] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000.

[CL01] J. Camenisch and A. Lysyanskaya. "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation". In: *Advances in Cryptology — EUROCRYPT 2001*. Ed. by B. Pfitzmann. Vol. 2045. LNCS. Springer Berlin Heidelberg, 2001, pp. 93–118.

Part I

Quantization using Jet Space Geometry

Chapter 1

The Schouten Bracket

The Schouten bracket (or antibracket) plays a central role in the Poisson formalism, as well as in the Batalin-Vilkovisky quantization of gauge systems and the BV-Laplacian. In the latter case, the bracket measures the failure of the BV-Laplacian to be a derivation with respect to the product of integral functionals, and it distinguishes physical and non-physical quantum observables through the quantum master equation (see section 2.3 on p. 28 for more details).

There are several (in)equivalent ways to realize the Schouten bracket on jet spaces. After having introduced some basic concepts and notations in the fist section that we will use throughout the thesis, we will compare a number of possible definitions for the Schouten bracket, examining in what ways they agree or disagree and how they relate to the case of usual manifolds.

This chapter is based on the following article.

[KR12] A. V. Kiselev and S. Ringers. "A comparison of definitions for the Schouten bracket on jet spaces". In: *Proceedings of the Sixth International Workshop "Group Analysis of Differential Equations and Integrable Systems"*. Larnaca, Cyprus, 2012, 15p. arXiv: 1208.6196.

The initiative for this article came from Arthemy Kiselev. My contribution has been the text itself, as well as almost one half of the mathematical content, which was jointly developed by both authors.

1.1 The geometry of jet space

In this section we introduce some notations and review basic facts that we will use throughout this part of the thesis. For a more detailed exposition of these matters, see for example [Olv00; KV11; KV99; Vin01; Kis12c].

1.1.1 The infinite jet bundle

Let $\pi : E \to M$ be a vector bundle. Then the *infinite jet bundle* $J^\infty(\pi)$ is usually defined as the inverse direct limit of the sequence

$$\cdots \xrightarrow{\pi_{k+1,k}} J^k(\pi) \xrightarrow{\pi_{k,k-1}} J^{k-1}(\pi) \xrightarrow{\pi_{k-1,k-2}} \cdots \xrightarrow{\pi_{2,1}} J^1(\pi) \xrightarrow{\pi_{1,0}} J^0(\pi) = E \xrightarrow{\pi} M.$$

More concretely, an element of $J^\infty(\pi)$ consists of a sequence $\theta = (x, \theta_k)_{k\in\mathbb{N}}$ such that

- $\theta_k \in J^k(\pi)$ for all $k \in \mathbb{N}$,
- if $k > l$ then $\pi_{k,l}(\theta_k) = \theta_l$.

Then there are projections $\pi_\infty : J^\infty(\pi) \to M$ and $\pi_{\infty,k} : J^\infty(\pi) \to J^k(\pi)$, defined respectively by $\pi_\infty(\theta) = x$ and $\pi_{\infty,k}(\theta) = \theta_k$.

We organize the coordinates as follows. x^i are the coordinates, with indices $i, j, k, \dots$, along the base manifold; q^α are the fiber coordinates of the bundle E, with indices $\alpha, \beta, \gamma \dots$. A point from the infinite jet space is then $\theta = (x^i, q^\alpha, q^\alpha_{x^i}, q^\alpha_{x^i x^j}, \dots, q^\alpha_\sigma, \dots) \in J^\infty(\pi)$, where σ is a multi-index. If $s \in \Gamma(\pi)$ is a section of π we denote with $j^\infty(s)$ its *infinite jet*, which is a section $j^\infty(s) \in \Gamma(\pi_\infty)$. Its value at $x \in M$ is $j^\infty_x(s) = (x^i, s^\alpha(x), \frac{\partial s^\alpha}{\partial x^i}(x), \dots, \frac{\partial^{|\sigma|} s^\alpha}{\partial x^\sigma}(x), \dots) \in J^\infty(\pi)$.

The following theorem has important consequences for our purposes.

Theorem 1.1 (Borel's lemma). *For any infinite sequence of real numbers $(a_0, a_1, \dots) \in \mathbb{R}^\infty$ there exists a smooth function $f : \mathbb{R} \to \mathbb{R}$ that has a_k as its k-th Tailor coefficients; that is, the Tailor series of f at zero is*[1]

$$\sum_{k=0}^{\infty} \frac{1}{k!} a_k x^k.$$

After a lot of generalizations this theorem has the following consequence.

Corollary 1.2. *If $\pi : E \to M$ is a vector bundle and $\theta \in J^\infty(\pi)$ is such that $\pi_\infty(\theta) = x$ for some $x \in M$, then θ is the jet at x of some smooth local section of π. That is, there exists a local section $s : U \to E$ (where $U \subset M$) such that $\theta = j^\infty_x(s)$.*

Because of this, we may interpret elements $\theta \in J^\infty(\pi)$ as the Taylor coefficients of local sections.

We define the ring of smooth functions $\mathcal{F}(\pi)$ on $J^\infty(\pi)$ as

$$\mathcal{F}(\pi) = \{f : J^\infty(\pi) \to \mathbb{R} \mid \exists k \in \mathbb{N} \text{ such that } f \in C^\infty(J^k(\pi))\} \cup C^\infty(M).$$

Thus, smooth functions on $J^\infty(\pi)$ are required to depend on only finitely many coordinates. Sometimes we will also write $C^\infty(J^\infty(\pi))$ for $\mathcal{F}(\pi)$.

[1] Notice that this theorem does not include any requirement on the convergence of the series. Therefore, it is entirely possible that this expansion has a convergence radius of 0 – it may converge only at $x = 0$ (at which it *does* converge, as is easily seen). Even worse, since the function f that is claimed to exist is only smooth and not nessecarily analytic, if this series *does* converge outside of 0 then it need not equal f at those points. For example, consider the smooth function f that has $f(x \neq 0) = \exp(-1/x^2)$ and $f(0) = 0$. All of its derivatives are 0 at 0, so its Tailor series at 0 does not agree with f at any point except 0.

1.1.2 The Einstein summation convention

Throughout this thesis we shall employ the *Einstein summation convention*: when an index variable appears twice in a single term, once as an upper and once as a lower index, it implies summation of that term over all the values of the index. For example, if ω is a one-form on some manifold M with coordinates x^i, then we will write $\omega = \omega_i \, \mathrm{d}x^i$; the summation $\sum_{i=1}^{n}$ is implied. We employ the following additional conventions.

1. When an upper index appears on a coordinate located under the bar in a derivative, as in $\frac{\partial}{\partial x^i}$, then the i counts as a *lower* index. Similarly, in $\frac{\partial}{\partial b_\alpha}$ the α counts as an upper index. This lets us write, for example, vector fields as $X = X^i \frac{\partial}{\partial x^i}$.
2. If σ is a multi-index and it appears twice in a term, then an infinite summation over all multi-indices is implied.
3. The exception to these rules are when the index i or multi-index σ appears as the exponent of a sign. When it does, it does not imply a summation. This allows us to write the variational derivative concisely as[2] $\frac{\delta}{\delta q^\alpha} = (-)^\sigma D_\sigma \frac{\partial}{\partial q^\alpha_\sigma}$ (note that there is still a summation over all multi-indices implied by the other two occurrences of σ).

Whenever there is reason to deviate from these rules we will explicitly write the summation signs.

1.1.3 Tangent vectors and vector fields

If $\theta = j^\infty_x(s) \in J^\infty(\pi)$ is an arbitrary element of the infinite jet space $J^\infty(\pi)$, then the *total derivatives* at that point are the vector fields D_i defined by

$$D_i\big|_{j^\infty_x(s)} := j^\infty(s)_* \left.\frac{\partial}{\partial x^i}\right|_x$$

which in coordinates is

$$D_i = \frac{\partial}{\partial x^i} + q^\alpha_{\sigma+1_i} \frac{\partial}{\partial q^\alpha_\sigma}. \tag{1.1}$$

In addition, if $\sigma = (i_1, \ldots, i_n)$ is a multi-index we define $D_\sigma = (D_1)^{i_1} \circ (D_2)^{i_2} \circ \cdots \circ (D_n)^{i_n}$.

Since vertical tangent vectors on $J^\infty(\pi)$ are spanned by the tangent vectors $\frac{\partial}{\partial q^\alpha_\sigma}$, it is immediate that the total derivatives form an Ehresmann connection, called the *Cartan connection* $\mathcal{C}$. Moreover, since they are defined as pushforwards, it is also immediate that the total derivatives commute, i.e. they form a involutive distribution on $J^\infty(\pi)$, and the Cartan connection is flat. Thus, the tangent space $T_\theta J^\infty(\pi)$ splits: $T_\theta J^\infty(\pi) = \mathcal{C}_\theta \oplus V_\theta$, where $V_\theta := \ker(D_\theta \pi_\infty)$ is the vertical subspace of $T_\theta J^\infty(\pi)$.

We say that a vector field X on $J^\infty(\pi)$ is *evolutionary* when both $[X, \mathcal{C}] \subset \mathcal{C}$ and X is vertical, i.e., $X \in \ker(\pi_\infty)_*$. Solving these requirements yields that any evolutionary

[2] We write $(-)^n$ as an abbreviation for $(-1)^n$, and when σ is a multi-index we write $(-)^\sigma$ instead of $(-)^{|\sigma|}$.

vector field is of the form

$$\partial_\varphi^{(q)} = D_\sigma(\varphi^\alpha)\frac{\partial}{\partial q^\alpha_\sigma},$$

where $\varphi^\alpha \in \mathcal{F}(\pi)$. We call $\varphi = (\varphi^1, \dots, \varphi^\alpha)$ the *generating section* of $\partial_\varphi^{(q)}$; by construction this is a section of the pullback bundle $\pi^*_\infty(\pi)\colon \pi^*_\infty(E) \to J^\infty(\pi)$. Denoting $\varkappa(\pi) := \Gamma(\pi^*_\infty(\pi)) = \Gamma(\pi) \otimes_{C^\infty(M)} \mathcal{F}(\pi)$ for brevity, then, we thus have that variational vector fields are in one-to-one correspondence with elements $\varphi \in \varkappa(\pi)$ by $\varphi \mapsto \partial_\varphi^{(q)}$, and we will often use this correspondence to identify the two concepts. We also transfer the bracket of vector fields to $\varkappa(\pi)$ by setting

$$[\partial_{\varphi_1}^{(q)}, \partial_{\varphi_2}^{(q)}] =: \partial_{[\varphi_1,\varphi_2]}^{(q)},$$

from which it follows that $[\varphi_1, \varphi_2] = \partial_{\varphi_1}^{(q)}(\varphi_2) - \partial_{\varphi_2}^{(q)}(\varphi_1)$.

In the remainder we will often just write vector for evolutionary vector fields.

1.1.4 Differential forms and covectors

We denote the space of all k-forms on $J^\infty(\pi)$ by $\Lambda^k(\pi)$. We say that a differential form $\omega \in \Lambda^k(\pi)$ is *horizontal* if it yields zero when any of its arguments is vertical; that is, $X \lrcorner\, \omega = 0$ for any vertical vector field X. We denote the space of horizontal k-forms by $\overline{\Lambda}^k(\pi)$. As vertical vector fields are of the form $X^\alpha_\sigma \frac{\partial}{\partial q^\alpha_\sigma}$, it follows that a form is horizontal only if it can be written as $\omega_{i_1,\dots,i_k}\, \mathrm{d}x^{i_1} \wedge \dots \wedge \mathrm{d}x^{i_k}$. Similarly, we say that a form is *vertical* when it yields zero whenever one or more of its arguments is horizontal. Thus the set of differential k-forms splits into a horizontal and a vertical part:

$$\Lambda^k(\pi) = \bigoplus_{i=0}^{k} \overline{\Lambda}^i(\pi) \otimes \mathcal{C}^{k-i}\Lambda(\pi),$$

and we call $\mathcal{C}^{k-i}\Lambda(\pi)$ the space of *vertical forms*.

This splitting induces a similar splitting of the de Rham differential $\mathrm{d}_{\mathrm{dR}(J^\infty(\pi))}$ on $J^\infty(\pi)$ into a horizontal part $\overline{\mathrm{d}}$ and a vertical part $\mathrm{d}_\mathcal{C}$:

$$\mathrm{d}_{\mathrm{dR}(J^\infty(\pi))} = \overline{\mathrm{d}} + \mathrm{d}_\mathcal{C},$$

where $\overline{\mathrm{d}}\colon \overline{\Lambda}^i(\pi) \otimes \mathcal{C}^j\Lambda(\pi) \to \overline{\Lambda}^{i+1}(\pi) \otimes \mathcal{C}^j\Lambda(\pi)$ and $\mathrm{d}_\mathcal{C}\colon \overline{\Lambda}^i(\pi) \otimes \mathcal{C}^j\Lambda(\pi) \to \overline{\Lambda}^i(\pi) \otimes \mathcal{C}^{j+1}\Lambda(\pi)$. We call these operators the *horizontal* and *Cartan differentials*, respectively. They are indeed differentials, they anticommute with each other as well as with the de Rham differential $\mathrm{d}_{\mathrm{dR}(J^\infty(\pi))}$, and they are antiderivations of the wedge product. From

this it follows that in coordinates, these operators are

$$\overline{\mathrm{d}} = \mathrm{d}x^i \wedge D_i \qquad \text{and} \qquad \mathrm{d}_{\mathcal{C}} = \mathrm{d}_{\mathcal{C}} q^a_\sigma \wedge \frac{\partial}{\partial q^a_\sigma},$$

with the understanding that the vector fields on the right hand side of the wedge products act on their arguments by the Lie derivative, and where

$$\mathrm{d}_{\mathcal{C}} q^a_\sigma = \mathrm{d}q^a_\sigma - q^a_{\sigma+1_i} \mathrm{d}x^i.$$

Taking the space $\overline{\Lambda}^i \otimes \mathcal{C}^j\Lambda$ (dropping the suffix (π) for now), we consider the cohomology with respect to the horizontal differential $\overline{\mathrm{d}}$:

$$E_1^{i,j}(\pi) := \frac{\ker\left(\overline{\mathrm{d}}\colon \overline{\Lambda}^i \otimes \mathcal{C}^j\Lambda \to \overline{\Lambda}^{i+1} \otimes \mathcal{C}^j\Lambda\right)}{\operatorname{im}\left(\overline{\mathrm{d}}\colon \overline{\Lambda}^{i-1} \otimes \mathcal{C}^j\Lambda \to \overline{\Lambda}^{i} \otimes \mathcal{C}^j\Lambda\right)}$$

This is called the *$\mathcal{C}$-spectral sequence*. In addition, we set $\overline{H}^i(\pi) = E_1^{i,0}(\pi)$, that is, horizontal i-forms modulo the horizontal differential $\overline{\mathrm{d}}$. For $i = n = \dim(M)$ elements of the space $\overline{H}^n(\pi)$ are called Hamiltonians or Lagrangians. Denoting the equivalence class of a horizontal n-form $h\,\mathrm{d}^n x \in \overline{\Lambda}^n(\pi)$ by $\int h\,\mathrm{d}^n x$ (where $h \in \mathcal{F}(\pi)$ is a function), an element $\mathcal{H} = \int h\,\mathrm{d}^n x \in \overline{H}^n(\pi)$ maps a section $s \in \Gamma(\pi)$ to the reals as follows:

$$s \mapsto \mathcal{H}(s) = \int h(j^\infty_x(s))\,\mathrm{d}^n x.$$

If $\xi\colon F \to J^\infty(\pi)$ is a vector bundle and $P = \Gamma(\xi)$ its space of sections, then the *adjoint module* $\widehat{P}$ of P is defined to be the space $\operatorname{Hom}_{\mathcal{F}(\pi)}(P, \overline{\Lambda}^n(\pi))$, and we denote the coupling between a module P and its adjoint by $\langle\,,\rangle$. Applying this to $\varkappa(\pi) = \Gamma(\pi^*_\infty(\pi))$, we define the space of *covectors* to be $\widehat{\varkappa}(\pi) := \widehat{\varkappa(\pi)} = \operatorname{Hom}_{\mathcal{F}(\pi)}(\varkappa(\pi), \overline{\Lambda}^n(\pi))$. This space then satisfies $\widehat{\varkappa}(\pi) \cong (\mathcal{C}^1\Lambda(\pi) \otimes \overline{\Lambda}^n(\pi))/\operatorname{im}\overline{\mathrm{d}} = E_1^{n,1}(\pi)$. Elements of the right hand side of this isomorphism act on evolutionary vector fields $\partial^{(p)}_\varphi$ via the unique representative of the equivalence class that can be written as $p_\alpha\,\mathrm{d}_{\mathcal{C}} q^\alpha \wedge \mathrm{d}^n x$ for certain functions $p_\alpha \in \mathcal{F}(\pi)$. In other words, if $\varphi \in \varkappa(\pi)$ and $p = p_\alpha\,\mathrm{d}_{\mathcal{C}} q^\alpha \wedge \mathrm{d}^n x \in \widehat{\varkappa}(\pi)$, then $\langle p, \varphi\rangle = p(\partial^{(q)}_\varphi) = p_\alpha \varphi^\alpha\,\mathrm{d}^n x \in \overline{\Lambda}^n(\pi)$.

The *variational derivative* of a function $h \in \mathcal{F}(\pi)$ is defined by

$$\frac{\delta h}{\delta q^\alpha} = (-)^\sigma D_\sigma \frac{\partial h}{\partial q^\alpha_\sigma}. \tag{1.2}$$

In addition, we set $\delta = \mathrm{pr}_{\overline{H}^n(\pi)} \circ \mathrm{d}_{\mathcal{C}}$, i.e. $\delta\omega = \int \mathrm{d}_{\mathcal{C}}\omega \in \widehat{\varkappa}(\pi)$ for $\omega \in \overline{\Lambda}^n(\pi)$, which we also call the variational derivative, or the *Euler operator*. Then it follows that if

$\omega = h\,\mathrm{d}^n x \in \overline{\Lambda}^n(\pi)$ we have

$$\delta\omega = \int \frac{\delta h}{\delta q^\alpha}\,\mathrm{d}_C q^\alpha \wedge \mathrm{d}^n x =: \int \frac{\delta h}{\delta q^\alpha}\delta q^\alpha; \tag{1.3}$$

the notation defined in the right hand side is slightly abusive (as strictly speaking $\delta q^\alpha \in \overline{H}^n(\pi)$) but convenient. Sometimes we need to keep track with respect to what coordinates q the Euler operator takes its variational derivatives; in that case we will write δ_q. Finally, it is easy to see that the variational derivative of functions of the form $D_i h$ is zero. As a consequence, if $\xi \in \overline{\Lambda}^{n-1}(\pi)$ then $\delta(\overline{\mathrm{d}}\xi) = 0$. Therefore, the action of δ descends to cohomology, $\delta\colon \overline{H}^n(\pi) \to \widehat{\varkappa}(\pi)$, so that we may write $\delta\mathcal{H} = \int \delta(h\,\mathrm{d}^n x)$.

1.1.5 Horizontal jet bundles

Let ξ be a vector bundle over $J^\infty(\pi)$, and suppose s_1 and s_2 are two sections of this bundle. We say that they are *horizontally equivalent* [KKV04] at a point $\theta \in J^\infty(\pi)$ if $D_\sigma(s_1^\alpha) = D_\sigma(s_2^\alpha)$ at θ for all multi-indices σ and fiber-indices α. Denote the equivalence class by $[s]_\theta$. The set

$$\overline{J^\infty_\pi}(\xi) := \{[s]_\theta \mid s \in \Gamma(\xi),\, \theta \in J^\infty(\pi)\}$$

is called the *horizontal jet bundle* of ξ. It is clearly a bundle over $J^\infty(\pi)$, whose elements above θ are determined by all the derivatives $s^\alpha_\sigma := D_\sigma(s^\alpha)$ for all multi-indices σ and fiber-indices α.

Now consider this formalism applied to bundles over $J^\infty(\pi)$ of the form $\pi^*_\infty(\zeta)$. Thus now ζ is a bundle over M that we pull back via π^*_∞ to create a bundle over $J^\infty(\pi)$; the construction above then takes jet space of this bundle (with respect to the horizontal coordinates). Alternatively, one might wonder what happens if we do this the other way around, i.e. we first take the jet space of ζ, obtaining a bundle $J^\infty(\zeta) \xrightarrow{\zeta_\infty} M$, and then pull back along π^*_∞. The following proposition shows that these two operations commute, while simultaneously providing a convenient description of the resulting jet space. We make the following convention: if π and ζ are the projections of two bundles over the same manifold M, then we denote the projection of the Whitney sum of π and ζ by $\pi \times_M \zeta$.

Proposition 1.3. *As bundles over $J^\infty(\pi)$, the horizontal jet bundle $\overline{J^\infty_\pi}(\pi^*_\infty(\zeta))$ and $\pi^*_\infty(J^\infty(\zeta))$ are equivalent. Moreover, as a bundle over M they are both equivalent to the Whitney sum $J^\infty(\pi) \times_M J^\infty(\zeta) = J^\infty(\pi \times_M \zeta)$.*

Proof. The pullback bundle $\pi^*_\infty(J^\infty(\zeta))$ is as a set equal to

$$\pi^*_\infty(J^\infty(\zeta)) = \{(j^\infty_x(q), j^\infty_x(u)) \in J^\infty(\pi) \times J^\infty(\zeta) \mid x \in M\},$$

from which we immediately see that $\pi^*_\infty(J^\infty(\zeta))$ is equivalent to the Whitney sum $J^\infty(\pi \times_M \zeta)$ of π and ζ. Thus it remains to show that $\pi^*_\infty(J^\infty(\zeta))$ and $\overline{J^\infty_\pi}(\pi^*_\infty(\zeta))$ are equivalent as bundles over $J^\infty(\pi)$.

Consider an element $[s]_\theta \in \overline{J^\infty_\pi}(\pi^*_\infty(\zeta))$. Thus, s is a section $s \in \Gamma(\pi^*_\infty(\zeta))$. By Borel's theorem (see Corollary 1.2 on page 4), an arbitrary element over $x \in M$ from $J^\infty(\pi)$ can be written as $j^\infty_x(q)$ for some $q \in \Gamma(\pi)$. Now define a section $u \in \Gamma(\zeta)$ by $u := j^\infty(q)^* s = s \circ j^\infty(q)$, i.e., $u(x) = s(j^\infty_x(q))$. Then by the definition of the total derivative, we have

$$\frac{\partial u^a}{\partial x^i}(x) = \frac{\partial}{\partial x^i}\big(s^a \circ j^\infty(q)\big)(x) = \big(D_{x^i} s^a\big)\big(j^\infty_x(q)\big),$$

that is, the partial derivatives of u and the total derivatives of s coincide. This shows that if we define a map by

$$[s]_{j^\infty_x(q)} \mapsto (j^\infty_x(q), j^\infty_x(u)) \in \pi^*_\infty(J^\infty(\zeta)),$$

where u is the section associated to s and q as outlined above, then this map is well-defined and smooth. Moreover, since the partial derivatives of a section u at x and the total derivatives of a section s at $j^\infty_x(q)$ completely define the equivalence classes $j^\infty_x(u)$ and $[s]_{j^\infty_x(q)}$ respectively, this map is also a bijection. Lastly, it is clear that as a bundle morphism over $J^\infty(\pi)$, it preserves fibers. □

When ζ is a bundle over M instead of over $J^\infty(\pi)$, and there is no confusion possible, we will abbreviate $\overline{J^\infty_\pi}(\pi^*_\infty(\zeta))$ with $\overline{J^\infty_\pi}(\zeta)$.

This identification endows the horizontal jet space $\overline{J^\infty_\pi}(\zeta)$ with the Cartan connection – namely the pullback connection on $\pi^*_\infty(J^\infty(\zeta))$. Therefore there exist total derivatives D_i on the horizontal jet space $\overline{J^\infty_\pi}(\zeta)$; in coordinates these are just, denoting the fiber coordinate of ζ with u^β, the operators

$$D_i = \frac{\partial}{\partial x^i} + q^\alpha_{\sigma+1_i}\frac{\partial}{\partial q^\alpha_\sigma} + u^\beta_{\tau+1_i}\frac{\partial}{\partial u^\beta_\tau}.$$

Thus, instead of the horizontal derivatives $D_\sigma(u^\alpha)$ of sections there are now the fiber coordinates u^α_σ, which have no derivatives along the fiber coordinates: $\frac{\partial}{\partial q^\alpha_\sigma} u^\beta_\tau = 0$.

1.1.6 Adjoint modules and total differential operators

Let $\xi : E \to J^\infty(\pi)$ be a vector bundle, and $P = \Gamma(\xi)$ its space of sections, which is an $\mathcal{F}(\pi)$-module. The *adjoint* module $\widehat{P}$ of P is defined to be the space

$$\widehat{P} := \mathrm{Hom}_{\mathcal{F}(\pi)}(P, \overline{\Lambda}^n(\pi)).$$

We denote the natural coupling between $\widehat{P}$ and P by $\langle\cdot,\cdot\rangle : \widehat{P} \times P \to \overline{\Lambda}^n(\pi)$.

Proposition 1.4. *If $\xi : E_\xi \to J^\infty(\pi)$ is a vector bundle over $J^\infty(\pi)$, then $\widehat{\Gamma(\xi)} = \Gamma(E^*_\xi \otimes \overline{\Lambda}^n(\pi))$.*

Furthermore, if $\zeta : E_\zeta \to M$ is a vector bundle over M, then, denoting $\widehat{\zeta} := E^*_\zeta \otimes \Lambda^n(M)$,

we have

$$\widehat{\Gamma(\pi_\infty^*(\zeta))} = \Gamma(\pi_\infty^*(\widehat{\zeta})).$$

Proof. We start with the first claim. Denote $P := \Gamma(\xi)$, let $p \in \widehat{P}$, and let $(e_i)_{i=1,\dots,k}$ be a frame for E_ξ (i.e. a set of sections e_i so that at each $\theta \in J^\infty(\pi)$, $(e_i(\theta))_{i=1,\dots,k}$ spans the fiber above θ), so that if $\varphi \in P$ then $\varphi = \varphi^i e_i$ for some functions $\varphi^i \in \mathcal{F}(\pi)$. Then $\langle p, \varphi\rangle = \langle p, \varphi^i e_i\rangle = \varphi^i \langle p, e_i\rangle$. But the coupling $\langle\cdot,\cdot\rangle$ gives an element of $\overline{\Lambda}^n(\pi)$, which is itself a space of sections, so it takes an argument θ from $J^\infty(\pi)$, and we see that

$$\langle p, \varphi\rangle(\theta) = \varphi^i(\theta) \cdot \langle p, e_i\rangle(\theta).$$

Therefore, $\langle p, \varphi\rangle(\theta)$ depends only on the value $\varphi(\theta)$ of φ at θ, and not on all of φ. Thus, if we define

$$\begin{aligned} &q(\theta)\colon \xi^{-1}(\theta) \subset E_\xi \to \overline{\Lambda}^n(\pi), \\ &q(\theta)(\varphi(\theta)) := \langle p, \varphi\rangle(\theta) \end{aligned}$$

then we see that this definition makes sense, and that $q(\theta)$ acts linearly on elements of $\xi^{-1}(\theta)$ and returns a number times $\mathrm{d}^n x$. That is, $q(\theta)$ is an element of the fiber of $E_\xi^* \otimes \overline{\Lambda}^n(\pi)$ above θ, which acts on $\xi^{-1}(\theta)$ by letting the first component of the tensor product act. q itself, then, is a smooth section $\Gamma(E_\xi^* \otimes \overline{\Lambda}^n(\pi))$. This space of sections already has an obvious $\mathcal{F}(\pi)$-linear action on P by letting the first component of the tensor product act pointwise on elements of P, so that we have found a correspondence that goes both ways.

If, moreover, $\xi = \pi_\infty^*(\zeta)$ for some vector bundle $\zeta : E_\zeta \to M$, then[3]

$$\begin{aligned} \widehat{P} = \widehat{\Gamma(\pi_\infty^*(\zeta))} &= \Gamma\Big(\pi_\infty^*(\zeta)^* \otimes \overline{\Lambda}^n(\pi)\Big) \\ &= \Gamma\Big(\pi_\infty^*(\zeta^*) \otimes \pi_\infty^*(\Lambda^n(M))\Big) \\ &= \Gamma\Big(\pi_\infty^*(E_\zeta^* \otimes \Lambda^n(M))\Big) = \Gamma(\pi_\infty^*(\widehat{\xi})). \end{aligned}$$

□

Let ξ, ξ' be two vector bundles over $J^\infty(\pi)$. Set $P_\xi := \Gamma(\xi)$ and similarly $P_{\xi'}$. An $\mathbb{R}$-linear map $\Delta : P_\xi \to P_{\xi'}$ is called a *total differential operator* if for any point $\theta \in J^\infty(\pi)$ and any section $q \in P_\xi$ the value of $\Delta(q)$ at θ is completely determined by the values of $D_\sigma(q)$ at θ. For the space of these operators we write $\mathcal{C}\mathrm{Diff}(P_\xi, P_{\xi'})$. For the space of maps that take k arguments and is linear and $\mathcal{C}$-differential in each of them, we write $\mathcal{C}\mathrm{Diff}_k(P_\xi, P_{\xi'})$.

[3] Alternatively, recall that $P := \pi_\infty^*(\zeta) = \Gamma(\zeta) \otimes_{C^\infty(M)} \mathcal{F}(\pi)$. Take a pure tensor from this space, $s \otimes f$ with $s \in \Gamma(\zeta)$ and $f \in \mathcal{F}(\pi)$. Also take some $p \in \widehat{P}$. Then we have $p(s \otimes f) = p(f \cdot (s \otimes 1)) = f p(s \otimes 1)$. That is, p is completely determined by how it acts on s, so we may regard it as an element $p \in \mathrm{Hom}_{C^\infty(M)}(\Gamma(\zeta), \Lambda^n(M))$. Now via an identical argument as in the first part of this proof, $p(s)(x)$ in fact only depends on $s(x)$ and not on all of s so that it comes from some $q \in E_\zeta^* \otimes \Lambda^n(M)$.

In coordinates, a $\Delta \in \mathcal{C}\mathrm{Diff}(P_\xi, P_{\xi'})$ is a matrix of the form

$$(\Delta)_{ab} = \sum_\sigma a_\sigma^{ab} D_\sigma,$$

where $a = 1, \dots, \dim \xi$, $b = 1, \dots, \dim(\xi')$ and $a_\sigma^{ab} \in \mathcal{F}(\pi)$.

Note that any map Δ of this form can also be seen as a linear bundle morphism $\Delta : \overline{J^\infty_\pi}(\xi) \to \xi'$, by setting $\Delta([q]_\theta) = \Delta(q)(\theta)$.

Definition 1.5. If Q is another module of sections of another bundle over $J^\infty(\pi)$ and $\Delta : P \to Q$ is a total differential operator, then the *adjoint* $\Delta^\dagger : \widehat{Q} \to \widehat{P}$ of Δ is defined as the map that satisfies for all $\hat{q} \in \widehat{Q}$ and $p \in P$

$$\int \langle \hat{q}, \Delta(p) \rangle = \int \langle \Delta^\dagger(\hat{q}), p \rangle.$$

In other words, $\Delta^\dagger$ is such that $\langle q, \Delta(p) \rangle - \langle \Delta^\dagger(q), p \rangle$ is $\overline{\mathrm{d}}$-exact.

1.2 The Schouten bracket

The Schouten bracket is a natural generalization of the commutator of vector fields to the fields of multivectors. It was introduced by J. A. Schouten [Sch40; Sch53], who with A. Nijenhuis [Nij55] established its main properties. Later it was observed by A. Lichnerowicz [Lic77; Lic78], that the bracket provides a way to check if a bivector π on a manifold determines a Poisson bracket via the formula $[\![\pi, \pi]\!] = 0$, which was the first intrinsically coordinate-free method to see this and established the use of the bracket in the Poisson formalism. Moreover, this makes the bracket instrumental in the definition of the Poisson(-Lichnerowich) cohomology on a Poisson manifold.

Historically, the bracket on jet space seems to have been researched in two distinct areas of mathematics and physics, which have been separate for a long time. The first branch is the quantization of gauge systems; here the bracket is known as the . It occurs for example in the seminal papers on the BRST and BV formalism, [BRS75; BRS76; Tyu75] and [BV81; BV83] respectively, where it is used to create a nilpotent operator $[\![\Omega, \cdot]\!]$ providing a resolution of the space of observables. Other occurences of the bracket in this context are [Zin75], [CF00], [HT92] and [Wit90], the last of which contains some geometrical interpretation of the bracket.

In the Poisson formalism on jet spaces it was understood in [GD80] that the bracket plays a similar role for recognizing Poisson brackets as on usual manifolds. Concepts such as Hamiltonian operators and the relation of the bracket with the Yang-Baxter equation are developed in [GD80] and [RS94; RS03]; for a review, see [Dor93]. A version of the bracket that can be restricted to equations was developed in [KKV04]. Later, a different, recursive way of defining the bracket, that we will discuss in this chapter, was shown in [KV11].

Generalizations to the $\mathbb{Z}_2$-setup and the purely non-commutative setting of the entire theory have been discussed in [Kon93], [OS98], and more recently [Kis12b]; for a review,

see [Kis12c].

The realization that the brackets in these areas of mathematics and physics coincide is not an obvious one. Accordingly, a number of seemingly distinct ways of defining the bracket has been developed, of which the equivalence is not always immediate and sometimes a subtle issue. In the remainder of this chapter we will examine four of those definitions, of which three will turn out to be equivalent when care is taken.

The remainder of this chapter is structured as follows. In section 1.3 the notion of variational multivectors (which will be the arguments of the bracket) are introduced; at this point it will become clear why the definition of the bracket for usual manifold fails in the case of jet spaces. In section 1.4 we first define the Schouten bracket as an odd Poisson bracket; then, after giving some examples of the bracket acting on two multivectors, we show that this definition is equivalent to the recursive one introduced in [KV11]. Using the recursive definition we shall prove the Jacobi identity for the bracket, which yields a third definition for the bracket, in terms of graded vector fields and their commutators.

1.3 Variational multivectors

In the case of a smooth manifold M, one can think of multivectors in two distinct but equivalent ways:

- A multivector of degree k is a (sum of) wedge products of k vector fields, where the wedge product is defined as the (normalized) antisymmetrization of the tensor product.
- A multivector of degree k is a map $\omega\colon (\Lambda^1(M))^k \to C^\infty(M)$ that is linear over $C^\infty(M)$ and completely antisymmetric.

The natural pairing between the set of vector fields and $\Lambda^1(M)$ can be extended to a pairing $\langle\,,\rangle\colon (\bigwedge^i TM) \times \Lambda^n(M) \to C^\infty(M)$, and this pairing then provides an identification between these two viewpoints.

For reasons that will become clear later on (see Remark 1.11 on p. 15) the first viewpoint is not available to us when we want to generalize the concept of multivectors to the variational setup. Therefore we will take the second. To that end, consider the bundle $\widehat{\pi} : E^* \otimes \Lambda^n(M) \to M$. Then $\pi_\infty^*(\widehat{\pi}) = \pi_\infty^*(E^*) \otimes \overline{\Lambda}(\pi)$, so that $\widehat{\varkappa}(\pi) = \Gamma(\pi_\infty^*(\widehat{\pi}))$. Thus, the formalism of horizontal jet spaces (see section 1.1.5 on p. 8) is applicable to covectors, so we either take p to be an element from $\widehat{\varkappa}(\pi)$, an actual covector, or $p \in \overline{J^\infty_\pi}(\widehat{\pi})$.

Definition 1.6. A *variational k-vector* ω, or just a k-vector, is a $\mathcal{F}(\pi)$-linear function that maps k covectors into $\overline{H}^n(\pi)$, i.e.,

$$\omega\colon \left(\widehat{\varkappa}(\pi)\right)^k \to \overline{H}^n(\pi),$$

that is totally skew-symmetric, and moreover a *total differential operator*: that is, its value is completely determined by the total derivatives of its arguments.[4] The number $k = |\omega|$

[4] Here we mean *all* total derivatives D_σ - including the one that has empty index $\sigma = \varnothing$, which is the identity. That is, ω is also allowed to depend on its arguments directly.

is called its *degree*.

Note that for arbitrary ω (i.e., not necessarily skew-symmetric), by partial integration we can always write $\omega(p_1,\dots,p_k) = \int\langle p_1, A(p_2,\dots,p_k)\rangle$, for some operator A, which then is a $(k-1)$-linear map taking values in $\varkappa(\pi)$. The demand that ω be totally skew-symmetric then implies firstly that A is also totally skew-symmetric, and secondly that it is *skew-adjoint* in each of its arguments:

$$\begin{aligned}\omega(p_1,\dots,p_k) &= \int\langle p_1, A(p_2,\dots,p_k)\rangle\\ &= -\int\langle p_j, A(p_1,\dots,p_{j-1},p_1,p_{j+1},\dots,p_k)\rangle.\end{aligned}$$

Let us write a k-vector ω as $\omega(p_1,\dots,p_k) = \int f(p_1,\dots,p_k)\,\mathrm{d}^n x$ for some function f. This function f is linear in each of its arguments, but as the next example shows, f need not be totally antisymmetric.

Example 1.7. Consider the following operator, which is associated to the Korteweg-De Vries-equation: $A_2^{\mathrm{KdV}} = -\frac{1}{2}D_3 + 2qD + q_x$, or equivalently $\langle p^1, A_2^{\mathrm{KdV}}(p^2)\rangle = \left(-\frac{1}{2}p^1p^2_{xxx} + 2qp^1p^2_x + q_xp^1p^2\right)\mathrm{d}x$. This is not antisymmetric because none of the terms are antisymmetric. However, we can fix this non-antisymmetry by integrating by parts:

$$\begin{aligned}\langle p^1, A_2^{\mathrm{KdV}}(p^2)\rangle &\cong \left(-\frac{1}{4}(p^1p^2_{xxx} + p^1p^2_{xxx}) + 2qp^1p^2_x - (qp^1_xp^2 + qp^1p^2_x)\right)\mathrm{d}x\\ &\cong \left(\frac{1}{4}p^1_xp^2_{xx} - \frac{1}{4}p^1_{xx}p^2_x + qp^1p^2_x - qp^1_xp^2\right)\mathrm{d}x,\end{aligned}$$

which is now explicitly antisymmetric.

We now make the observations above more formal.

Remark 1.8. Let ω be a k-vector. Then there exists a function $f\colon (\widehat{\varkappa}(\pi))^k \to \mathcal{F}(\pi)$ in total derivatives, which is totally antisymmetric, such that

$$\omega(p_1,\dots,p_k) = \int f(p_1,\dots,p_k)\,\mathrm{d}^n x.$$

Indeed, if $\omega(p_1,\dots,p_k) = \int g(p_1,\dots,p_k)\,\mathrm{d}^n x$ for some non-antisymmetric function g, then since ω is totally antisymmetric, we have

$$\begin{aligned}\omega(p_1,\dots,p_k) &= \frac{1}{k!}\sum_{\sigma\in S_k}(-)^\sigma\omega(p_{\sigma(1)},\dots,p_{\sigma(k)})\\ &= \int\left(\frac{1}{k!}\sum_{\sigma\in S_k}(-)^\sigma g(p_{\sigma(1)},\dots,p_{\sigma(k)})\right)\mathrm{d}^n x,\end{aligned}$$

so $f(p_1,\dots,p_k) := \frac{1}{k!}\sum_{\sigma\in S_k}(-)^\sigma g(p_{\sigma(1)},\dots,p_{\sigma(k)})$ is such a function. Note, however, that f is not unique, because constructing f in this way for the example above would

yield an expression which differs from the one in the example.

At this point we take the fibers of the bundle $\widehat{\pi}$ and of $\pi_\infty^*(\widehat{\pi})$, and reverse their parity, $\Pi\colon p \mapsto b$, while we keep the entire underlying jet space intact [Vor02]. The result is the horizontal jet space $\overline{J^\infty_\pi}(\Pi\widehat{\pi})$ with odd fibers over x. An element θ from this space has coordinates

$$\theta = (x^i, q^\alpha, q^\alpha_{x^i}, \dots, q^\alpha_\sigma, \dots; b_\alpha, b_{\alpha,x^i}, \dots, b_{\alpha,\sigma}, \dots).$$

The coupling $\langle p, \varphi\rangle = p_\alpha \varphi^\alpha \,\mathrm{d}^n x$ extends tautologically to the odd b's, as do the total derivatives: $D_\sigma b_\alpha = b_{\alpha,\sigma}$.

Definition 1.9. A *noncommutative variational k-vector*, with $k \in \mathbb{N} \cup \{0\}$, is an element of $\overline{H}^n(\pi_\infty^*(\Pi\widehat{\pi}))$ having a density that is k-linear in the odd b's or their derivatives (i.e., it is a homogeneous polynomials of degree k in $b_{\alpha,\sigma}$). If ξ is a k-vector we will call $k = \deg(\xi) = |\xi|$ its *degree*. Note that by partial integration, any such k-vector ξ can be written as

$$\xi(b) = \int \langle b, A(b, \dots, b)\rangle$$

for some total totally skew-symmetric total differential operator A that takes $k-1$ arguments, takes values in $\varkappa(\pi)$, and is skew-adjoint in each of its arguments (e.g., in the case of a 2-vector, $\int \langle b^1, A(b^2)\rangle = \int \langle b^2, A(b^1)\rangle$). We will also use the term *variational multivector* if the vector is not necessarily homogeneous, or if we do not want to specify the degree of the variational multivector.

Note further that this does not imply that *every* density is, or has to be, a homogeneous polynomial of degree k; for example, $\int bb_x \,\mathrm{d}^n x = \int (bb_x + D_x(bb_x b_{xx}))\,\mathrm{d}^n x$.

Suppose we have some coordinate expression for a representative of the (nonzero) cohomology class $\omega(b)$. Then each term contains each jet coordinate b^a_σ at most once, because if such a coordinate would occur twice they would square to zero. Therefore, it is possible to see from which slot the jet coordinate came from. This allows us to *evaluate* the noncommutative k-vector on k covectors: we set

$$\omega(p_1, \dots, p_k) = \frac{1}{k!} \sum_{\sigma\in S_k} (-)^\sigma \omega(p_{\sigma(1)}, \dots, p_{\sigma(k)}), \tag{1.4}$$

i.e. in the coordinate expression of (the representative of) ω we replace the i-th b that we come across with $p_{\sigma(i)}$ (moving from left to right), and sum over all permutations $\sigma \in S_k$. It is clear that the resulting expression is completely antisymmetric in its arguments, and is an element of $\overline{H}^n(\pi)$; that is, it is a k-vector in the sense of Definition 1.6. In fact, the following is now clear.

Proposition 1.10. *Let ω be a k-vector. The map that assings to ω the noncommutative k-vector defined by $\omega(b) := \omega(b, \dots, b)$ for $b \in \overline{J^\infty_\pi}(\Pi\widehat{\pi})$ is an injection, whose injective inverse is given by equation (1.4).*

Thus, the notions of k-vectors and noncommutative k-vectors coincide, and henceforth we will treat them as the same. This machinery allows us to give a particularly convenient form of the variational Schouten bracket.

Remark 1.11. Contrary to the case of usual manifolds M, where the space of k-vectors is isomorphic to $\bigwedge^k TM$, the space of variational k-vectors does not split in such a fashion. As a result, the two formulas

$$[\![X, Y \wedge Z]\!] = [\![X, Y]\!] \wedge Z + (-)^{(|X|-1)|Y|} Y \wedge [\![X, Z]\!] \tag{1.5}$$

for multivectors X, Y and Z, and

$$\begin{aligned} &[\![X_1 \wedge \cdots \wedge X_k, Y_1 \wedge \cdots \wedge Y_\ell]\!] \\ &\quad = \sum_{\substack{i \le i \le k \\ 1 \le j \le \ell}} (-)^{i+j} [X_i, Y_j] \wedge X_1 \wedge \cdots \wedge \widehat{X_i} \wedge \cdots \wedge X_k \wedge Y_1 \wedge \cdots \wedge \widehat{Y_j} \wedge \cdots \wedge Y_\ell \end{aligned} \tag{1.6}$$

for vector fields X_i and Y_j, no longer hold. Both of these formulas provide a way of defining the bracket on usual, smooth manifolds (together with $[\![X, f]\!] = X(f)$ for vector fields X and functions $f \in C^\infty(M)$, and $[\![X, Y]\!] = [X, Y]$ for vector fields X and Y).

To sketch an argument why the space of variational k-vectors does not split in this way, take for example a 0-vector $\omega = \int f \, \mathrm{d}^n x$ and a 1-vector, which we can write as $\eta = \int \langle b, \varphi \rangle$ for some $\varphi \in \varkappa(\pi)$. How would we define the wedge product $\omega \wedge \eta$? Both of the factors contain a volume form and if we just put them together using the wedge product we get 0, so this approach does not work.

Suppose then we set in this case $\omega \wedge \eta = \int f \langle b, \varphi \rangle$. Now the problem is that f is not uniquely determined by ω and φ is not uniquely determined by η; both are fixed only up to $\overline{\mathrm{d}}$-exact terms. For example, $\omega = \int f \, \mathrm{d}^n x = \int (f + D_i(g)) \, \mathrm{d}^n x$, but $\int f \langle b, \varphi \rangle \neq \int f \langle b, \varphi \rangle + \int D_i(g) \langle b, \varphi \rangle$, because the second term is in general not identically zero.

Similarly, we have $\eta = \int \langle b, \varphi \rangle = \int \big(\langle b, \varphi \rangle + \overline{\mathrm{d}}(\alpha(b)) \big)$, for any linear map α mapping b into $(n-1)$-forms. In the same way as above, this trivial term stops being trivial whenever we multiply it on the left with the density of a 0-vector, say. The difficulty persists for multivectors of any degree k and so there is no reasonable wedge product or splitting. (We will, however, solve this absence more or less by hands in Chapter 2, by Definition 2.11 on p. 36.)

1.4 Definitions of the bracket

1.4.1 Odd Poisson bracket

Now that there are odd coordinates b_α, we have to distinguish between *left* and *right derivatives*. Suppose $f \in \mathcal{F}(\pi \times \Pi\widehat{\pi})$ is a function on $J^\infty(\pi \times \Pi\widehat{\pi})$ that does not depend on $b_{\alpha,\sigma}$ for some indices α, σ. Then the left and right derivatives with respect to $b_{\alpha,\sigma}$ are

defined respectively by

$$\frac{\overrightarrow{\partial}}{\partial b_{\alpha,\sigma}}(b_{\alpha,\sigma}f) = f, \qquad \frac{\overleftarrow{\partial}}{\partial b_{\alpha,\sigma}}(fb_{\alpha,\sigma}) = f.$$

For arbitrary $f, g \in \mathcal{F}(\pi \times \Pi\widehat{\pi})$ these imply the following graded Leibniz rules:

$$\frac{\overrightarrow{\partial}}{\partial b_{\alpha,\sigma}}(fg) = \frac{\overrightarrow{\partial}}{\partial b_{\alpha,\sigma}}(f)g + (-)^{|f|}f\frac{\overrightarrow{\partial}}{\partial b_{\alpha,\sigma}}g$$
$$\frac{\overleftarrow{\partial}}{\partial b_{\alpha,\sigma}}(fg) = f\frac{\overleftarrow{\partial}}{\partial b_{\alpha,\sigma}}g + (-)^{|g|}\frac{\overleftarrow{\partial}}{\partial b_{\alpha,\sigma}}(f)g.$$

Thus, when operating on a product of functions the arrow indicates on which factor the derivative acts first. As for the variational derivatives, we set (c.f. equation (1.2))

$$\frac{\overrightarrow{\delta}}{\delta b_\alpha} = (-)^\sigma D_\sigma \frac{\overrightarrow{\partial}}{\partial b_{\alpha,\sigma}} \qquad \text{and} \qquad \frac{\overleftarrow{\delta}}{\delta b_\alpha} = (-)^\sigma D_\sigma \frac{\overleftarrow{\partial}}{\partial b_{\alpha,\sigma}},$$

so that the relation between the Euler operator δ_b and the left and right variational derivatives is (c.f. equation (1.3))

$$\delta_b \omega = \int \delta b_\alpha \frac{\overrightarrow{\delta\omega}}{\delta b_\alpha} = \int \frac{\overleftarrow{\delta\omega}}{\delta b_\alpha}\delta b_\alpha,$$

that is, the arrow on top of the variational derivative points away from the variation δb_α.

Definition 1.12. Let ξ and η be variational multivectors respectively. The *variational Schouten bracket* $[\![\xi, \eta]\!]$ of ξ and η is the $(|\xi| + |\eta| - 1)$-vector defined by[5]

$$[\![\xi, \eta]\!] = \int \left[\frac{\overleftarrow{\delta\xi}}{\delta q^\alpha}\frac{\overrightarrow{\delta\eta}}{\delta b_\alpha} - \frac{\overleftarrow{\delta\xi}}{\delta b_\alpha}\frac{\overrightarrow{\delta\eta}}{\delta q^\alpha} \right] \tag{1.7}$$

in which one easily recognizes a Poisson bracket. The fact that this is a $(|\xi| + |\eta| - 1)$-vector comes from the variational derivatives $\frac{\delta}{\delta b}$ occuring in the expression: if η takes $|\eta|$ arguments then $\frac{\delta\eta}{\delta b}$ takes $|\eta| - 1$ arguments.

We will use the following two lemmas to calculate examples 1.15 through 1.18.

[5]To be precise, if $\xi = \int f(b, \ldots, b)\, \mathrm{d}^n x$ and $\eta = \int g(b, \ldots, b)\, \mathrm{d}^n x$, where f and g are both homogeneous polynomials in $b_{\alpha,\sigma}$ of degree $|\xi|$ and $|\eta|$ respectively, then the bracket is given by

$$[\![\xi, \eta]\!] = \int \left(\frac{\overleftarrow{\delta f}}{\delta q^\alpha}\frac{\overrightarrow{\delta g}}{\delta b_\alpha} - \frac{\overleftarrow{\delta f}}{\delta b_\alpha}\frac{\overrightarrow{\delta g}}{\delta q^\alpha} \right) \mathrm{d}^n x,$$

which does not depend on the representatives f and g because $\delta \circ \overline{\mathrm{d}} = 0$. This notation, although correct, does not seem to be used in the literature.

Lemma 1.13. *Suppose $\xi = \langle b, A(b,\dots,b)\rangle$ is a multivector. Then*

$$\frac{\overrightarrow{\delta\xi}}{\delta b_\alpha} = |\xi| A(b,\dots,b)^\alpha. \tag{1.8}$$

Proof. Set $k = |\xi|$. We calculate

$$\begin{aligned}\delta b_\alpha \frac{\overrightarrow{\delta\xi}}{\delta b_\alpha} = \delta_b \xi &= \delta_b \langle b, A(b,\dots,b)\rangle \\ &= \langle \delta b, A(b,\dots,b)\rangle + \sum_{n=1}^{k-1} \langle b, A(b,\dots,\delta b,\dots,b)\rangle\end{aligned}$$

Now δb anticommutes with the b left to it, and A is antisymmetric in all of its arguments, so we can switch δb with the b on its left, giving two cancelling minus signs. Doing this multiple times, we obtain

$$= \langle \delta b, A(b,\dots,b)\rangle + \sum_{j=1}^{k-1} \langle b, A(\delta b, b,\dots,b)\rangle.$$

Next we first switch δb with the b to its left, and then use the fact that A is skew-symmetric in its first argument, again giving two cancelling minus signs:

$$\begin{aligned}&= \langle \delta b, A(b,\dots,b)\rangle + (k-1)\langle \delta b, A(b,\dots,b)\rangle \\ &= k\langle \delta b, A(b,\dots,b)\rangle \\ &= k\, \delta b_\alpha A(b,\dots,b)^\alpha.\end{aligned}$$

The result follows by comparing the coefficients of δb^α. □

Lemma 1.14. *Let ξ and η be k and ℓ-vectors respectively, so that $\xi = \int \langle b, A(b,\dots,b)\rangle$ and $\eta = \int \langle b, B(b,\dots,b)\rangle$ respectively. Then*

$$[\![\xi,\eta]\!] = \int \left[(-)^{k(\ell-1)}\ell \partial^{(q)}_{B(b,\dots,b)}\xi - (-)^{k-1} k \partial^{(q)}_{A(b,\dots,b)}\eta\right]. \tag{1.9}$$

Proof. In the second term of the definition of the Schouten bracket, we first reverse the arrow on the b-derivative, giving a sign $(-)^{k-1}$. In the first term, we swap the two factors $(\overleftarrow{\delta\xi}/\delta q^\alpha)(\overrightarrow{\delta\eta}/\delta b_\alpha)$. For this we have to move the $\ell-1$ b's of $\overrightarrow{\delta\eta}/\delta b_\alpha$ through the k b's of $\overleftarrow{\delta\xi}/\delta q^\alpha$, giving a sign $(-)^{k(\ell-1)}$. Thus

$$[\![\xi,\eta]\!] = \int \left[\frac{\overleftarrow{\delta\xi}}{\delta q^\alpha}\frac{\overrightarrow{\delta\eta}}{\delta b_\alpha} - \frac{\overleftarrow{\delta\xi}}{\delta b_\alpha}\frac{\overrightarrow{\delta\eta}}{\delta q^\alpha}\right]$$

$$= \int \left[(-)^{k(\ell-1)} \frac{\overrightarrow{\delta\eta}}{\delta b_\alpha} \frac{\overleftarrow{\delta\xi}}{\delta q^\alpha} - (-)^{k-1} \frac{\overrightarrow{\delta\xi}}{\delta b_\alpha} \frac{\overrightarrow{\delta\eta}}{\delta q^\alpha} \right]$$
$$= \int \left[(-)^{k(\ell-1)} \ell D_\sigma B(b)^\alpha \frac{\partial \xi}{\partial q^\alpha_\sigma} - (-)^{k-1} k D_\sigma A(b)^\alpha \frac{\partial \eta}{\partial q^\alpha_\sigma} \right]$$
$$= \int \left[(-)^{k(\ell-1)} \ell \partial^{(q)}_{B(b)} \xi - (-)^{k-1} k \partial^{(q)}_{A(b)} \eta \right]. \qquad \square$$

Example 1.15. Let $\mathcal{H} \in \overline{H}^n(\pi)$ be a 0-vector, and take a one-vector $\varphi \in \varkappa(\pi)$, i.e., $\eta = \langle b, \varphi \rangle$. As $k = 0$ the term on the right in equation (1.9) vanishes. We are left with

$$[\![\mathcal{H}, \varphi]\!] = \int \partial^{(q)}_\varphi \mathcal{H},$$

i.e., the Schouten bracket calculates the velocity of $\mathcal{H}$ along $\partial^{(q)}_\varphi$.

Example 1.16. Suppose ξ and η are two one-vectors, i.e., $\xi = \int \langle b, \varphi_1 \rangle$ and $\eta = \int \langle b, \varphi_2 \rangle$ for some $\varphi_1, \varphi_2 \in \varkappa(\pi)$. Then

$$[\![\xi, \eta]\!] = \int \left(\partial^{(q)}_{\varphi_2}(\xi) - \partial^{(q)}_{\varphi_1}(\eta) \right) = \int \left(\partial^{(q)}_{\varphi_2} \langle b, \varphi_1 \rangle - \partial^{(q)}_{\varphi_1} \langle b, \varphi_2 \rangle \right)$$
$$= \int \left(\langle b, \partial^{(q)}_{\varphi_2} \varphi_1 \rangle - \langle b, \partial^{(q)}_{\varphi_1} \varphi_2 \rangle \right)$$

which holds because b does not depend on the jet coordinates q^α, whence

$$= \int \langle b, [\varphi_2, \varphi_1] \rangle = - \int \langle b, [\varphi_1, \varphi_2] \rangle.$$

Thus, in this case the variational Schouten bracket just calculates the ordinary commutator of evolutionary vector fields, up to a minus sign (c.f. equation (1.6)).

Example 1.17. Suppose the base and fiber are both $\mathbb{R}$, and let $\xi = \int b b_x \, \mathrm{d}x$ be a (nontrivial) two-vector and $\eta = \int b x^3 q_{xx} \, \mathrm{d}x$ be a one-vector. Then, again using equation (1.9),

$$[\![\xi, \eta]\!] = 0 + \int 2 \partial^{(q)}_{b_x} (b x^3 q_{xx}) \, \mathrm{d}x = 2 \int D_x^2(b_x) b x^3 \, \mathrm{d}x = 2 \int x^3 b_{xxx} b \, \mathrm{d}x.$$

We shall return to this example on p. 21 (see Example 1.23).

Example 1.18. In this final example, let $\xi = \int b b_x \, \mathrm{d}x$ again and $\eta = \int q_x b b_x \, \mathrm{d}x$; then

$$[\![\xi, \eta]\!] = 0 + 2 \int \partial^{(q)}_{b_x} (q_x b b_x) \, \mathrm{d}x = 2 \int D_x(b_x) \cdot b b_x \, \mathrm{d}x = 2 \int b b_x b_{xx} \, \mathrm{d}x.$$

Notice the factor 2 standing in front of the answers in the last two examples; it will become important in the next section.

1.4.2 A recursive definition

The second way of defining the bracket, due to I. Krasil'shchik and A. Verbovetsky [KV11], is done in terms of the *insertion operator*: let ξ be a k-vector, and let $p \in \widehat{\varkappa}(\pi)$ or $p \in \overline{J^\infty_\pi}(\widehat{\pi})$ (i.e., p can be either an actual covector or an element from the corresponding horizontal jet space). Denote by $\xi(p)$ or $\iota_p(\xi)$ the $(k-1)$-vector that one obtains by putting p in the rightmost slot of ξ:

$$\xi(p)(b) = \iota_p(\xi)(b) = \xi(\underbrace{b,\dots,b}_{k-1},p) = \frac{1}{k}\sum_{j=1}^{k}(-)^{k-j}\xi(b,\dots,b,p,b,\dots,b), \tag{1.10}$$

where p is in the j-th slot in the right hand side. Note that if we were to insert $k-1$ additional elements of $\widehat{\varkappa}(\pi)$ in this expression in this way, we recover formula (1.4).

Lemma 1.19. *If $\xi = \int \langle b, A(b,\dots,b)\rangle$ is a k-vector, then*

$$\frac{\overrightarrow{\delta\xi}(p)}{\delta b_\alpha} = \frac{k-1}{k}\frac{\overrightarrow{\delta\xi}}{\delta b_\alpha}(p) \quad \textit{and} \quad \frac{\overleftarrow{\delta\xi}(p)}{\delta b_\alpha} = -\frac{k-1}{k}\frac{\overleftarrow{\delta\xi}}{\delta b_\alpha}(p). \tag{1.11}$$

Proof. $\xi(p)$ is a $(k-1)$-vector, so $\overrightarrow{\delta\xi}(p)/\delta b_\alpha = (k-1)A(b,\dots,b,p)^\alpha$ by Lemma 1.13. However, ξ is a k-vector, so $(\overrightarrow{\delta\xi}/\delta b_\alpha)(p) = (kA(b,\dots,b)^\alpha)(p) = kA(b,\dots,b,p)^\alpha$, from which the first equality of the Lemma follows. The second equality is established by reversing the arrow of the derivative, using the first equality, and restoring the arrow to its original direction again; this results in the extra minus sign in this equality. □

On the other hand, if $p \in \overline{J^\infty_\pi}(\widehat{\pi})$ then $\delta\xi(p)/\delta q^\alpha = (\delta\xi/\delta q^\alpha)(p)$. Indeed, we have $\partial p_{\beta,\tau}/\partial q^\alpha_\sigma = 0$, and if f is one of the densities of a k-vector, then the total derivative D_i and the insertion operator ι_p commute. For example,

$$\iota_p(D_i b_{\alpha,\sigma}) = \iota_p(b_{\alpha,\sigma+1_i}) = p_{\alpha,\sigma+1_i}$$

and

$$D_i(\iota_p(b_{\alpha,\sigma})) = D_i(p_{\alpha,\sigma}) = p_{\alpha,\sigma+1_i}.$$

Thus, from the formula $\frac{\delta}{\delta q^\alpha} = \sum_{|\sigma|>0}(-)^\sigma D_\sigma \frac{\partial}{\partial q^\alpha_\sigma}$ for the variational derivative it follows that $\delta\xi(p)/\delta q^\alpha = (\delta\xi/\delta q^\alpha)(p)$.

Theorem 1.20. *Let ξ and η be k and ℓ-vectors, respectively, and $p \in \overline{J^\infty_\pi}(\widehat{\pi})$. Then*

$$[\![\xi,\eta]\!](p) = \frac{\ell}{k+\ell-1}[\![\xi,\eta(p)]\!] + (-)^{\ell-1}\frac{k}{k+\ell-1}[\![\xi(p),\eta]\!]. \tag{1.12}$$

Proof. We relate the two sides of the equation by letting p range over the slots as in equation (1.10). In this calculation we will for brevity omit the fiber indices α.

Consider the first term of the left hand side, $\left(\frac{\overleftarrow{\delta\xi}}{\delta q}\frac{\overrightarrow{\delta\eta}}{\delta b}\right)(p)$. If we were to take the sum as in equation (1.10), we would obtain an expression containing $k+\ell-1$ slots; in some cases p is in one of the $\ell-1$ slots of $\overrightarrow{\delta\eta}/\delta b$ and in the other cases it is in one of the k slots of $\overleftarrow{\delta\xi}/\delta q$. All of these terms carry the normalizing factor $1/(k+\ell-1)$. Now we notice the following:

- Each term in which p is in a slot coming from $\overrightarrow{\delta\eta}/\delta b$ has a matching term in the expansion of $\frac{\overleftarrow{\delta\xi}}{\delta q}\iota_p\left(\frac{\overrightarrow{\delta\eta}}{\delta b}\right)$ according to (1.10), except that there each term would carry a factor $1/(\ell-1)$, because now p only has access to the $\ell-1$ slots of $\overrightarrow{\delta\eta}/\delta b$.
- Similarly, each term of the left hand side of (1.12) in which p is in one of the slots of $\overleftarrow{\delta\xi}/\delta q$ has a matching term in the expansion of $\iota_p\left(\frac{\overleftarrow{\delta\xi}}{\delta q}\right)\frac{\overrightarrow{\delta\eta}}{\delta b}$, but there they carry a factor $1/k$.
- Moreover, in that case they also carry the sign $(-)^{\ell-1}$, which comes from the fact that here p had to pass over the $\ell-1$ slots of $\overrightarrow{\delta\eta}/\delta b$.

Gathering these remarks, we find

$$\begin{aligned}\left(\frac{\overleftarrow{\delta\xi}}{\delta q}\frac{\overrightarrow{\delta\eta}}{\delta b}\right)(p) &= \frac{\ell-1}{k+\ell-1}\frac{\overleftarrow{\delta\xi}}{\delta q}\iota_p\left(\frac{\overrightarrow{\delta\eta}}{\delta b}\right)+(-)^{\ell-1}\frac{k}{k+\ell-1}\iota_p\left(\frac{\overleftarrow{\delta\xi}}{\delta q}\right)\frac{\overrightarrow{\delta\eta}}{\delta b}\\ &= \frac{\ell}{k+\ell-1}\frac{\overleftarrow{\delta\xi}}{\delta q}\frac{\overrightarrow{\delta\eta}(p)}{\delta b}+(-)^{\ell-1}\frac{k}{k+\ell-1}\frac{\overleftarrow{\delta\xi}(p)}{\delta q}\frac{\overrightarrow{\delta\eta}}{\delta b},\end{aligned}$$

where we have used the first equation of Lemma 1.19 in the first term.

Now we consider the second term of the left hand side of (1.12), and use a similar reasoning:

$$\begin{aligned}\left(\frac{\overleftarrow{\delta\xi}}{\delta b}\frac{\overrightarrow{\delta\eta}}{\delta q}\right)(p) &= \frac{\ell}{k+\ell-1}\frac{\overleftarrow{\delta\xi}}{\delta b}\iota_p\left(\frac{\overrightarrow{\delta\eta}}{\delta q}\right)+(-)^{\ell}\frac{k-1}{k+\ell-1}\iota_p\left(\frac{\overleftarrow{\delta\xi}}{\delta b}\right)\frac{\overrightarrow{\delta\eta}}{\delta q}\\ &= \frac{\ell}{k+\ell-1}\frac{\overleftarrow{\delta\xi}}{\delta b}\frac{\overrightarrow{\delta\eta}(p)}{\delta q}+(-)^{\ell+1}\frac{k}{k+\ell-1}\frac{\overleftarrow{\delta\xi}(p)}{\delta b}\frac{\overrightarrow{\delta\eta}}{\delta q},\end{aligned}$$

where now the second equation of Lemma 1.19 has been used. Subtracting the results of these two calculations, we obtain exactly the right hand side of equation (1.12). □

Thus, by recursively reducing the degrees of the arguments of the bracket, formula (1.12) expresses the value of the bracket of a k-vector and an ℓ-vector on $k+\ell-1$ covectors. We can interpret it as a second definition of the Schouten bracket, provided that we also set

$$[\![\mathcal{H},\varphi]\!]=\int\partial_\varphi^{(q)}\mathcal{H}=\int\langle\delta\mathcal{H},\varphi\rangle$$

for 1-vectors φ and 0-vectors $\mathcal{H} \in \overline{H}^n(\pi)$. Theorem 1.20 then says that this definition is equivalent to Definition 1.12. However, let us notice the following:

Remark 1.21. There are numerical factors in front of the two terms of the right hand side; these are absent in [KV11]. For example, the bracket of a 2-vector ξ and a 0-vector $\mathcal{H}$ is $[\![\mathcal{H}, \xi]\!](p) = 2\xi(\delta\mathcal{H}, p)$ according to both Definition 1.12 and Theorem 1.20; note the factor 2.

Remark 1.22. Secondly, it is important that the p that is inserted in (1.12) is *not* an actual covector, but that $p \in \overline{J^\infty_\pi}(\widehat{\pi})$. Otherwise, unwanted terms like $\partial^{(q)}_\varphi(p)$ occur in the final steps, and equivalence with Definition 1.12 is spoiled. Thus one takes two multivectors, inserts elements from the horizontal jet space according to the formula, and only plugs in the (derivatives of) actual covectors at the end of the day. This remark is again absent from [KV11].

Example 1.23. Let us re-calculate Example 1.17 using this formula. So, let $\xi = \int bb_x \,\mathrm{d}x$ and $\eta = \int bx^3 q_{xx} \,\mathrm{d}x$, and let $p^1, p^2 \in \overline{J^\infty_\pi}(\widehat{\pi})$. Then

$$\begin{aligned}
[\![\xi, \eta]\!](p^1, p^2) &= [\![\xi, \eta]\!](p^2)(p^1) = \frac{1}{2}[\![\xi, \eta(p^2)]\!](p^1) + \frac{2}{2}[\![\xi(p^2), \eta]\!](p^1) \\
&= -2 \cdot \frac{1}{2} \cdot [\![\xi(p^1), \eta(p^2)]\!] + 1 \cdot [\![\xi(p^2), \eta(p^1)]\!] + 1 \cdot [\![\xi(p^1, p^2), \eta]\!] \\
&= \int \Big[(-)^2 \, \partial^{(q)}_{p^1_x}(p^2 x^3 q_{xx}) - \partial^{(q)}_{p^2_x}(p^1 x^3 q_{xx}) + \tfrac{1}{2}\partial^{(q)}_{x^3 q_{xx}}(p^1 p^2_x - p^2 p^1_x) \Big] \,\mathrm{d}x \\
&= \int \Big[x^3 p^1_{xxx} p^2 - (p^1 \rightleftarrows p^2) \Big] \,\mathrm{d}x.
\end{aligned}$$

(Keeping track of the coefficients and signs is a good exercise.) This is precisely what one gets after evaluating the result of Example 1.17 on p^1 and p^2.

Theorem 1.20 allows us to reduce the Jacobi identity for the Schouten bracket to that of the commutator of one-vectors.

Proposition 1.24. *Let r, s and t be the degrees of the variational multivectors ξ, η and ζ, respectively. The Schouten bracket satisfies the graded Jacobi identity*:

$$(-)^{(r-1)(t-1)} [\![\xi, [\![\eta, \zeta]\!]]\!] + (-)^{(r-1)(s-1)} [\![\eta, [\![\zeta, \xi]\!]]\!] + (-)^{(s-1)(t-1)} [\![\zeta, [\![\xi, \eta]\!]]\!] = 0. \tag{1.13}$$

Proof. We proceed by induction using Theorem 1.20. When the degrees of the three vectors do not exceed 1, the statement follows from the reductions of the Schouten bracket to known structures, as in Examples 1.15 and 1.16. Now let the degrees be arbitrary natural numbers. Denote by I_1, I_2 and I_3 the respective terms of the left hand

side of (1.13). Then for any $p \in \overline{J^\infty_\pi}(\widehat{\pi})$ we have that

$$\begin{aligned}
I_1(p) &= (-)^{(r-1)(t-1)}[\![\xi,[\![\eta,\zeta]\!]]\!](p)\\
&= \frac{(-)^{(r-1)(t-1)}}{r+s+t-2}\Big((s+t-1)[\![\xi,[\![\eta,\zeta]\!](p)]\!]+r(-)^{s+t-2}[\![\xi(p),[\![\eta,\zeta]\!]]\!]\Big)\\
&= \frac{(-)^{(r-1)(t-1)}}{r+s+t-2}(t[\![\xi,[\![\eta,\zeta(p)]\!]]\!]+s(-)^{t-1}[\![\xi,[\![\eta(p),\zeta]\!]]\!]\\
&\qquad\qquad +r(-)^{s+t-2}[\![\xi(p),[\![\eta,\zeta]\!]]\!]\Big).
\end{aligned}$$

Similarly,

$$\begin{aligned}
I_2(p) &= \frac{(-)^{(r-1)(s-1)}}{r+s+t-2}(r[\![\eta,[\![\zeta,\xi(p)]\!]]\!]+t(-)^{r-1}[\![\eta,[\![\zeta(p),\xi]\!]]\!]\\
&\qquad\qquad +s(-)^{r+t-2}[\![\eta(p),[\![\zeta,\xi]\!]]\!]),\\
I_3(p) &= \frac{(-)^{(s-1)(t-1)}}{r+s+t-2}\Big(s[\![\zeta,[\![\xi,\eta(p)]\!]]\!]+r(-)^{s-1}[\![\zeta,[\![\xi(p),\eta]\!]]\!]\\
&\qquad\qquad +t(-)^{r+s-2}[\![\zeta(p),[\![\xi,\eta]\!]]\!]\Big).
\end{aligned}$$

For notational convenience, let us set $I_1(p)+I_2(p)+I_3(p) =: I/(r+s+t-2)$. Next we rearrange the terms in I:

$$\begin{aligned}
I = {}&(-)^{r-1}t\{(-)^{(r-1)(t-2)}[\![\xi,[\![\eta,\zeta(p)]\!]]\!]+(-)^{(r-1)(s-1)}[\![\eta,[\![\zeta(p),\xi]\!]]\!]\\
&+(-)^{(s-1)(t-2)}[\![\zeta(p),[\![\xi,\eta]\!]]\!]\}+(-)^{t-1}s\{(-)^{(r-1)(t-1)}[\![\xi,[\![\eta(p),\zeta]\!]]\!]\\
&+(-)^{(r-1)(s-2)}[\![\eta(p),[\![\zeta,\xi]\!]]\!]+(-)^{(s-2)(t-1)}[\![\zeta,[\![\xi,\eta(p)]\!]]\!]\}\\
&+(-)^{s-1}r\{(-)^{(r-2)(t-1)}[\![\xi(p),[\![\eta,\zeta]\!]]\!]+(-)^{(r-2)(s-1)}[\![\eta,[\![\zeta,\xi(p)]\!]]\!]\\
&+(-)^{(s-1)(t-1)}[\![\zeta,[\![\xi(p),\eta]\!]]\!]\},
\end{aligned}$$

i.e., we obtain the Jacobi identity for ξ, η and $\zeta(p)$; for ξ, $\eta(p)$ and ζ; and for $\xi(p)$, η and ζ (each times some unimportant factors). Thus we see that if we know that the identity holds for $(r-1,s,t)$, $(r,s-1,t)$ and $(r,s,t-1)$, then it holds for (r,s,t). □

1.4.3 Graded vector fields

Proposition 1.25. *If ξ and η are k and ℓ-vectors respectively, then their Schouten bracket is equal to*

$$[\![\xi,\eta]\!] = \int Q^\xi(\eta) = \int (\xi)\overleftarrow{Q}\eta, \tag{1.14}$$

where for any k-vector ξ, the graded evolutionary vector field Q^ξ is defined by

$$Q^\xi := \partial^{(q)}_{-\delta_b\xi} + \partial^{(b)}_{\delta_q\xi}. \tag{1.15}$$

Proof. This is readily seen from the equalities

$$\begin{aligned}\int \partial^{(b)}_{\delta_q\xi}(\eta) &= \int D_\sigma\left(\frac{\overleftarrow{\delta\xi}}{\delta q^\alpha}\right)\frac{\partial\eta}{\partial b_{\alpha,\sigma}} = \int \frac{\overleftarrow{\delta\xi}}{\delta q^\alpha}(-)^\sigma D_\sigma\frac{\partial\eta}{\partial b_{\alpha,\sigma}}\\ &= \int \frac{\overleftarrow{\delta\xi}}{\delta q^\alpha}\frac{\overrightarrow{\delta\eta}}{\delta b_\alpha}\end{aligned}$$

which is the first term of the Schouten bracket $[\![\xi,\eta]\!]$. The second term of (1.15) is done similarly. □

As a consequence of this proposition, the Schouten bracket is a derivation: if η is a product of k factors, then $[\![\xi,\eta]\!] = \int Q^\xi(\eta)$ has k terms, where in the i-th term, Q^ξ acts on the i-th factor while leaving the others alone. However, while the bracket is a derivation in both of its arguments separately, it is *not* a bi-derivation (i.e., a derivation in both arguments simultaneously), as in equation (1.6). To see why this is so, take a multivector η and let us suppose for simplicity that it has a density that consists of a single term containing ℓ coordinates, which can be either q's or b's: $\eta = \prod_{i=1}^{\ell} a_i$, for a set of letters a_i. Then the i-th term of $[\![\xi,\eta]\!] = \int Q^\xi(\eta)$ is a sign which is not important for the present purpose, times $a_1\cdots Q^\xi(a_i)\cdots a_\ell$.

Now suppose that $\xi = \prod_{j=1}^{k} c_j$ for some set of letters c_j, and note that $Q^\xi(a_i) = (\xi)\overleftarrow{Q}^{a_i}$ + trivial terms. Let us call the trivial term ω for the moment. Then we see that

$$a_1\cdots Q^\xi(a_i)\cdots a_\ell = a_1\cdots(\xi)\overleftarrow{Q}^{a_i}\cdots a_\ell + a_1\cdots\omega\cdots a_\ell.$$

Here the first term expands to what it should be in order for the bracket to be a bi-derivation, namely a sum consisting of terms of the form

$$a_1\cdots c_1\cdots(c_j)\overleftarrow{Q}^{a_i}\cdots c_k\cdots a_\ell$$

times possible minus signs. The second term, however, is generally no longer trivial, so that it does not vanish. Therefore the bracket is not in general a bi-derivation.

Theorem 1.26. *The Schouten bracket is related to the graded commutator of graded vector fields as follows*:

$$\int Q^{[\![\xi,\eta]\!]}f = \int [Q^\xi, Q^\eta]f, \tag{1.16}$$

for any smooth function f on the horizontal jet space $\overline{J^\infty_\pi}(\widehat{\pi})$.

Proof. Expanding the right hand side of equation (1.16) using the definition of the graded commutator and equation (1.15), we infer that

$$\begin{aligned}\int [Q^\xi, Q^\eta] f &= \int Q^\xi(Q^\eta(f)) - (-)^{(|\xi|-1)(|\eta|-1)} \int Q^\eta(Q^\xi(f)) \\ &= [\![\xi, [\![\eta, f]\!]]\!] - (-)^{(|\xi|-1)(|\eta|-1)} [\![\eta, [\![\xi, f]\!]]\!] = [\![[\![\xi, \eta]\!], f]\!] \\ &= \int Q^{[\![\xi,\eta]\!]} f.\end{aligned}$$

where we used the graded Jacobi identity (see Proposition 1.24) in the third line. □

This provides a third way of defining the Schouten bracket, equivalent to the previous two. Since the only fact that is used in this proof is that the Schouten bracket satisfies the graded Jacobi identity, Theorem 1.26 is actually equivalent to the Jacobi identity for the Schouten bracket. It is also possible to prove Theorem 1.26 directly (see [Kis12c, p. 84], by inspecting both sides of equation (1.16); in that case the Jacobi identity may be proved as a consequence of Theorem 1.26.

As a bonus, we see that if P is a Poisson bi-vector, i.e., $[\![P, P]\!] = 0$, then Q^P is a differential, $(Q^P)^2 = 0$. This gives rise to the Poisson(-Lichnerowicz) cohomology groups H^k_P.

Finally, we note that these definitions of the Schouten bracket also exist and remain coinciding in the $\mathbb{Z}_2$-graded setup $J^\infty\big((\pi_0|\pi_1)\big) \to M^{n_0|n_1}$, and in the setup of purely non-commutative manifolds and non-commutative bundles (see [Kon93], [OS98] and lastly [Kis12c], which contains details and discussion, and generalizes the topic of this chapter to the non-commutative world).

Chapter 2

The BV-formalism

In theoretical physics, the Batalin–Vilkovisky formalism is a technique for the quantization of certain gauge-invariant systems, formulated in terms of fields and anticommuting antifields; the Schouten bracket (usually called the antibracket); and a linear differential called the BV-Laplacian. Mathematically, the function(al)s on fields and antifields together with the Schouten bracket and the BV-Laplacian form a BV-algebra. Nearly all of this can be concisely formulated in terms of calculus on (parity-odd) jet bundles. Unfortunately, the BV-Laplacian as it is usually defined in the literature contains "infinite constants" or delta-functions that need to be regularized. In this chapter we will define the notion of BV-algebras and briefly explain their phyiscal importance, review and introduce the geometrical setup of such systems, and finally we explain why a particular Laplacian that contains no such infinite constants or delta functions is not suitable for use in the BV-formalism.

2.1 Introduction

The Batalin–Vilkovisky (BV) Laplacian Δ is a necessary ingredient in the quantization of gauge-invariant systems of Euler–Lagrange equations [BV81; BV83; HT92]. Its construction relies on the presence of canonically conjugate pairs of variables such as fields and antifields, or ghost-antighost pairs, which stem from the derivation of the Euler–Lagrange equations of motion $\delta S = 0$ and Noether relations, respectively (see [BV81; BV83; BRS75; BRS76; Tyu75] and [Kis12a; Kis12c]). The BV-Laplacian Δ is intimately connected to the variational Schouten bracket $[\![\,,]\!]$ (the odd Poisson bracket, or *antibracket* [Wit90]), which was discussed in the previous chapter.

When the Laplace equation $\Delta F = 0$ holds for an integral functional F of fields and antifields, then Feynman's path integral of F over the space of admissible fields is essentially independent of the non-physical antifields. As soon as the setup becomes

quantum and all objects depend also on the Planck constant $\hbar$, the Laplace equation $\Delta\big(\mathcal{O}\cdot\exp(iS^{\hbar}_{\mathrm{BV}}/\hbar)\big)=0$ selects the quantum observables $\mathcal{O}$ (here $S^{\hbar}_{\mathrm{BV}}$ is the extension in powers of $\hbar$ for the full BV-action $S_{\mathrm{BV}}=S+\ldots$ of a given gauge-invariant model $\delta S=0$). This approach yields the quantum master equation upon $S^{\hbar}_{\mathrm{BV}}$ and creates the cohomology groups with respect to the quantum BV-differential $\Omega=-i\hbar\Delta+[\![S^{\hbar}_{\mathrm{BV}},\cdot\,]\!]$. The observables are Ω-closed, $\Omega(\mathcal{O})=0$; Feynman's path integral is then used to calculate their expectation values and correlations.

On top of the difficulties which are immanent to a definition of the path integral, the BV-Laplacian Δ itself often suffers from a necessity to be regularized manually if one wishes to avoid the otherwise appearing "infinite constants" or Dirac's delta-distributions (e.g., see [CF00], in which such constants appear). On the other hand, in the case where the base space – that is, the space of the independent variables – reduces to a single point, the jet bundle becomes a supermanifold; in this case the BV-Laplacian contains no infinite constants [Sch93; Sch94; Sch99].

In this chapter we explore the geometrical setup and the physical importance of the Laplacian. Our considerations will revolve around the philosophy of *secondary calculus*, which, as explained in section 2.2, is the idea that when one reduces the base space to the zero-dimension manifold $\{\mathrm{pt}\}$, the tools and concepts at hand should reduce to known structures on (super)manifolds. Then, in section 2.3 we give the definition of BV-algebras, as well as briefly exploring the Laplacian on supermanifolds (as opposed to the Laplacian on jet bundles), giving a firm understanding of what the Laplacian looks like before its variational generalization. In sections 2.4 and 2.5 we introduce and extend the geometrical setup for the BV-Laplacian, after which we will examine a candidate BV-Laplacian on jet bundles in section 2.6.

2.2 Secondary calculus

Before we proceed it is convenient to briefly discuss the ideology of *secondary calculus*, which is a very instructive analogy between the geometry of jet spaces and that of smooth manifolds. This parallel was introduced by A. M. Vinogradov in [Vin01, Ch. 5]; see also [KV11; KV99; Vit09]. Without naming it we have already dealt with it in Chapter 1, where we generalized, for example, multivectors to variational multivectors and the Schouten bracket to its variational cousin; it will also play a role in both this chapter and Chapter 4.

The physics of a point particle can be succinctly and elegantly described by, for example, the Hamiltonian formalism on a smooth manifold E (see the introduction to Chapter 3 on p. 45). In this case its dynamics is specified by an ordinary differential equation, and expressed with the use of the calculus on the manifold E; that is, by differential forms, vector fields, the de Rham differential, et cetera. Note that the position of the point particle can be specified by an element $x\in E$, which are in one-to-one correspondence with sections of the trivial bundle $E\to\{\mathrm{pt}\}$. Thus, we have identified a bundle with a single fiber and trivial base space.

Now we consider two possible generalizations for such schemes (of which the first motivated the name "secondary calculus"):

1. The classical physics of point particles can be generalized to quantum mechanics; in this case the equation of motion (an ordinary differential equation) is replaced by the Schrödinger equation, which is a partial differential equation.
2. There are many other (classical) physical systems imaginable that do not involve point particles, but, for example, strings or fields. In these cases the dynamics are usually also described by a partial differential equation.

In both cases there is a bundle $\pi\colon E \to M$, inducing a jet bundle $J^\infty(\pi)$ in which the relevant partial differential equation lives. We see that with respect to the setup $E \to \{\mathrm{pt}\}$ of point particles described above, the base space $\{\mathrm{pt}\}$ has been replaced by a manifold M.

The idea of secondary calculus is that the calculus and machinery of infinite jet bundles may be used to describe the dynamics of these systems, in the same way as the dynamics of point particles can be expressed using calculus on manifolds. Thus, concepts such as smooth functions, vector fields, differential forms and the de Rham derivative all gain "secondary", or *variational*, generalizations. A guiding principle in the construction of these generalization is that they should reduce to their smooth counterparts when one takes the base space M of the bundle $E \xrightarrow{\pi} M$ to be the zero-dimensional manifold $\{\mathrm{pt}\}$.

Let us compile a brief list of such variational generalizations. As the position of a point particle is determined by a section $s\colon \{\mathrm{pt}\} \to M$, we find that the "points" in secondary calculus are the sections $s \in \Gamma(\pi)$. Having such a "point" and an element $\mathcal{H} = \int h\,\mathrm{d}^n x \in \overline{H}^n(\pi)$, we can calculate the number

$$\mathcal{H}(s) := \int_M h(j^\infty_x(s))\,\mathrm{d}^n x,$$

suggesting that $\overline{H}^n(\pi)$ is a natural generalization of the space of functions $C^\infty(E)$.[1] Referring also to Chapter 1, we may compile a dictionary of variational generalizations; see Table 2.1.

Manifold $E \xrightarrow{\pi} \{\mathbf{pt}\}$		**Jet space** $J^\infty(E \xrightarrow{\pi} M)$
points	$\longleftrightarrow$	sections of π
functions	$\longleftrightarrow$	elements $\int h\,\mathrm{d}^n x$ from $\overline{H}^n(\pi)$
value at a point, $f(x)$	$\longleftrightarrow$	integral $\int_M h(j^\infty_x(s))\,\mathrm{d}^n x$
vector fields	$\longleftrightarrow$	evolutionary vector fields, $\varkappa(\pi)$
multivector fields	$\longleftrightarrow$	variational multivectors
the de Rham complex	$\longleftrightarrow$	the $\mathcal{C}$-spectral sequence $E_1^{p,q}(\pi)$
de Rham differential d_{dR}	$\longleftrightarrow$	the Euler operator δ
the Schouten bracket	$\longleftrightarrow$	the variational Schouten bracket

Table 2.1. Variational generalizations of calculus on manifolds.

This chapter is essentially concerned with how the BV-Laplacian Δ fits into this table.

[1] Note, however, that under the contraction $M \mapsto \{\mathrm{pt}\}$ the space of functions $\mathcal{F}(\pi)$ on $J^\infty(\pi)$ also reduces to $C^\infty(E)$; this will become important in Chapter 4.

Although we are primarily interested in the entry on the right hand side, in the next section we shall examine what it looks like on (super)manifolds, as well as defining BV-algebras and briefly explaining their physical relevance.

2.3 BV-algebras and the quantum master equation

Definition 2.1. A *BV-algebra* A is a graded associative supercommutative algebra with unit, equipped with:

- A bilinear bracket $[\![\,,]\!]$ of order one satisfying a graded Leibniz rule with respect to the product of A, i.e. the bracket satisfies for all $x, y, z \in A$

$$\begin{aligned}
&[\![x,y]\!] = -(-)^{(|x|-1)(|y|-1)}[\![y,x]\!], && \text{(skew-symmetry)}\\
&[\![x,[\![y,z]\!]]\!] = [\![[\![x,y]\!],z]\!] + (-)^{(|x|-1)(|y|-1)}[\![y,[\![x,z]\!]]\!], && \text{(graded Jacobi identity)}\\
&[\![x,yz]\!] = [\![x,y]\!]z + (-)^{(|x|-1)|y|}y[\![x,z]\!]. && \text{(Leibniz rule)}
\end{aligned}$$

- A second order nilpotent differential operator Δ called the *Laplacian*, which has degree -1 with respect to the grading of A and satisfying

$$\begin{aligned}
&[\![x,y]\!] = (-)^{|x|}\left(\Delta(xy) - \Delta(x)y - (-)^{|x|}x\Delta y\right), && (2.1)\\
&\Delta^2 = 0,\\
&\Delta(1) = 0.
\end{aligned}$$

Thus the bracket measures the failure of the Laplacian being a left-derivation[2] with respect to the product of the algebra A. Moreover, we have the following immediate consequence.

Proposition 2.2. *The Laplacian is a derivation of the bracket,*

$$\Delta([\![x,y]\!]) = [\![\Delta x, y]\!] + (-)^{|x|-1}[\![x,\Delta y]\!]. \tag{2.2}$$

Proof. We calculate $\Delta([\![x,y]\!])$ by applying Δ to the right hand side of (2.1):

$$\begin{aligned}
\Delta([\![x,y]\!]) &= (-)^{|x|}\left(\Delta^2(xy) - \Delta(\Delta(x)y) - (-)^{|x|}\Delta(x\Delta y))\right)\\
&= (-)^{|x|-1}\left(\Delta(\Delta(x)y) + (-)^{|x|}\Delta(x\Delta y))\right). \qquad (2.3)
\end{aligned}$$

[2]With some adjustments to a number of minus signs, one may modify the Laplacian such that the bracket measures its failure in being a *right*-derivation. For example, in case of the Laplacian on a supermanifold (which we will discuss shortly), equation (2.5) would become $\Delta = \overleftarrow{\partial}/\partial x^i \circ \overleftarrow{\partial}/\partial\theta_i$ – that is, the arrows are reversed.

Next we expand the right hand side of (2.2) in the same way:

$$\begin{aligned}
&[\![\Delta x, y]\!] + (-)^{|x|-1}[\![x, \Delta y]\!] \\
&\quad = (-)^{|x|-1}\left(\Delta(\Delta(x)y) - \Delta^2(x)y - (-)^{|x|-1}\Delta x \Delta y\right) \\
&\qquad + (-)^{|x|-1}(-)^{|x|}\left(\Delta(x\Delta y) - \Delta x \Delta y - (-)^{|x|} x\Delta^2 y\right) \\
&\quad = (-)^{|x|-1}\left(\Delta(\Delta(x)y) + (-)^{|x|}\Delta(x\Delta y)\right).
\end{aligned}$$

This matches precisely with (2.3). □

As preparation for a first example of a BV-algebra, we first review some symplectic geometry. If M is a $2n$-dimensional symplectic manifold with symplectic form ω, then it has $\Omega = \omega^n$ as its natural volume form, and on any manifold having a volume form one can define the divergence $\operatorname{div}\colon \Gamma(TM) \to C^\infty(M)$ of a vector field by the following formula:

$$\operatorname{div}(X)\Omega = \mathrm{d}(i_X\Omega),$$

where i_X is the insertion operator that inserts the vector field X in the leftmost slot of its argument. Writing $\Omega = \rho\,\mathrm{d}^n x$ for the volume form, in a coordinate system x^i (not necessarily Darboux) this works out to

$$\operatorname{div} X = \rho^{-1}\partial_i(\rho X^i) = \partial_i X^i + X^i \partial_i \ln\rho.$$

(Note that ρ, being the density of a volume form, is nowhere zero.) Now the 2-form ω, being nondegenerate, has an inverse $\alpha = \omega^{-1} \in \bigwedge^2 TM$; that is, a 2-vector. One can use this 2-vector to define a Poisson bracket $\{\,,\}$ on $C^\infty(M)$ by the formula

$$\{f, g\} := \alpha(\mathrm{d}f, \mathrm{d}g).$$

Then, for any fixed function H we define its associated *Hamiltonian vector field* X_H by[3]

$$X_H = \{\cdot, H\} = \alpha(\mathrm{d}\cdot, \mathrm{d}H). \tag{2.4}$$

Now a Laplacian by definition has to be of degree -1 in the grading of its algebra, but $C^\infty(M)$ is an ungraded algebra. However, so far essentially none of the arguments above depend on that the coordinates x^i are even. Therefore, it is possible to generalize to the following [Sch93; Sch94; Sch99].

Example 2.3. A *P-manifold*, or a *symplectic supermanifold*, is a supermanifold of dimension

[3] Alternatively, one may define the Hamiltonian vector field uniquely by the formula

$$\mathrm{d}H = \omega(X_H, \cdot).$$

However, this way of doing this has the disadvantage that it directly uses ω, meaning that it only makes sense on symplectic manifolds. The definition given by (2.4) works for any Poisson manifold. We will also encounter it in Chapter 3, in which Poisson brackets play a central role.

$n|n$ equipped with a nondegenerate closed odd[4] 2-form ω. For such forms there is a super-Darboux theorem, stating that ω can always be written as $\omega = \mathrm{d}x^i \wedge \mathrm{d}\theta_i$ for a certain coordinate chart $(x^1, \dots, x^n, \theta_1, \dots, \theta_n)$.

When a P-manifold additionally has a form $\rho\, \mathrm{d}^n x\, \mathrm{d}^n\theta$ which is such that on each coordinate patch there exist Darboux coordinates such that locally $\rho(x, \theta) = 1$, then the supermanifold is said to be an *SP-manifold*.

Having this machinery in place, we can define (with due attention to the parity of the coordinates and resulting signs) the Hamiltonian vector field X_H of a function H and the divergence of a vector field X exactly as in the previous example. Using this machinery we finally define a Laplacian as follows:

$$\Delta f = \tfrac{1}{2} \operatorname{div} X_f.$$

Taking a Darboux coordinate chart[5] $(x^1, \dots, x^n, \theta_1, \dots, \theta_n)$, in coordinates these structures become

$$\begin{aligned} \{f, g\} &= \frac{\overleftarrow{\partial} f}{\partial x^i} \frac{\overrightarrow{\partial} g}{\partial \theta_i} - \frac{\overleftarrow{\partial} f}{\partial \theta_i} \frac{\overrightarrow{\partial} g}{\partial x^i}, \\ \Delta f &= \frac{\overrightarrow{\partial}}{\partial x^i} \frac{\overrightarrow{\partial}}{\partial \theta_i}. \end{aligned} \tag{2.5}$$

In order to verify that $(C^\infty(M), [\![\,,]\!], \Delta)$ is indeed a BV-algebra, all that is left to do is check that equation (2.1) holds, which is easily done using the expressions above.

Finally, we take $M = \Pi T^* E$ as our supermanifold. If x^i is a coordinate patch on E and θ_i are the induced parity-odd coordinates on $\Pi T^* E$, then we can express a natural symplectic odd 2-form on $\Pi T^* E$ by $\mathrm{d}x^i \wedge \mathrm{d}\theta_i$ as usual. Using the identification $C^\infty(\Pi T^* E) = \Gamma(\bigwedge^\bullet TE)$, we can transfer the Laplacian and the induced bracket to the space of multivectors on E. The bracket then precisely coincides with the Schouten bracket of multivectors on E. Referring to Table 2.1, this is the entry on the BV-Laplacian in the left column – meaning that we expect a variational generalization of the BV-Laplacian to be such that under the contraction $M \mapsto \{\mathrm{pt}\}$ we obtain this example.

Let us now briefly explain the primary physical application of BV-algebras. Thus, let $(A, [\![\,,]\!], \Delta)$ be a (complex) BV-algebra. In quantum field theory one is concerned with the Feynman path integral of elements of A (which in that case consists of (products of) integral functionals, i.e. $\mathfrak{M}(\pi_{\mathrm{BV}})$; this will be treated in detail in Section 2.5 on p. 36). Without precisely explaining what this path integral is, we will use the fact that the path integral of such a functional F can only be independent on the non-physical antifields when the following Laplace equation holds:

$$\Delta F = 0.$$

[4] If z^i with $i = 1, \dots, 2n$ is a coordinate patch and $|z_i| \in \mathbb{Z}/2\mathbb{Z}$ is the parity of these coordinates, then the parity of the two-form $\omega = \mathrm{d}z^i\, \omega_{ij}\, \mathrm{d}z^j$ is given by $|\omega| = |\omega_{ij}| + |z^i| + |z^j|$. For example, $\mathrm{d}x^i \wedge \mathrm{d}x^j$ is even, while $\theta^i\, \mathrm{d}\theta^j \wedge \mathrm{d}\theta^k$ and $\mathrm{d}x^i \wedge \mathrm{d}\theta_i$ are both odd.

[5] We write the index i of the odd coordinates θ_i as lowerscripts in order to be able to keep on using the Einstein summation convention (see section 1.1.2 on p. 5).

Adjoining the Planck constant and its inverse[6] to the BV-algebra A by $A^\hbar := A \otimes \mathcal{C}[\![\hbar, \hbar^{-1}]\!]$, the system is specified by the full BV-action $S^\hbar_{\mathrm{BV}} \in A^\hbar$, and the calculation of certain expectation values can be done by calculating the path integral over the functional $\exp(iS^\hbar_{\mathrm{BV}}/\hbar)$. For this reason the following will have to hold:

$$\Delta \exp\left(\frac{iS^\hbar_{\mathrm{BV}}}{\hbar}\right) = 0.$$

Moreover, an element $\mathcal{O} \in A^\hbar$ is called a *quantum observable* if the following equation holds:

$$\Delta\left(\mathcal{O} \exp\left(\frac{iS^\hbar_{\mathrm{BV}}}{\hbar}\right)\right) = 0.$$

These equations are our starting point; we will derive from them the quantum master equation (2.6) on $S^\hbar_{\mathrm{BV}}$ and equation (2.7) on the observable $\mathcal{O}$. (We refer to [HT92] for more details.)

Proposition 2.4. *Let $S^\hbar_{\mathrm{BV}} \in A^\hbar$ be even. If the identity $\Delta(\exp(iS^\hbar_{\mathrm{BV}}/\hbar)) = 0$ holds, then $S^\hbar_{\mathrm{BV}}$ satisfies the quantum master equation*:

$$\frac{1}{2}[\![S^\hbar_{\mathrm{BV}}, S^\hbar_{\mathrm{BV}}]\!] = i\hbar\Delta S^\hbar_{\mathrm{BV}}. \tag{2.6}$$

We will need the following two lemmas.

Lemma 2.5. *Let $F \in A^\hbar$ be even, let $G \in A^\hbar$ be another element, and let $n \in \mathbb{N}_{\geq 1}$. Then*

$$[\![G, F^n]\!] = n[\![G, F]\!]F^{n-1}.$$

Proof. We use induction on the Leibniz rule for the bracket $[\![\,,\,]\!]$. Note that all signs vanish since F is even, meaning that whenever F is multiplied with any element from $A^\hbar$, the factors may be freely swapped without this resulting in minus signs. For $n = 1$ the statement is trivial. Suppose the formula holds for some $n \in \mathbb{N}_{>1}$, then

$$\begin{aligned}[\![G, F^{n+1}]\!] &= [\![G, F \cdot F^n]\!] = [\![G, F]\!]F^n + F[\![G, F^n]\!] \\ &= [\![G, F]\!]F^n + nF[\![G, F]\!]F^{n-1} \\ &= (n+1)[\![G, F]\!]F^n,\end{aligned}$$

so that the statement also holds for $n + 1$. □

[6] Neither the Planck constant $\hbar$ nor the factor i is always present in articles on this subject; mainly papers that are more mathematical in nature sometimes skip one or both of these. Apparently, they do not always play a crucial role.

Lemma 2.6. *Let $F \in A^\hbar$ be even, and let $n \in \mathbb{N}_{\geq 2}$. Then*

$$\Delta(F^n) = n(\Delta F)F^{n-1} + \frac{1}{2}n(n-1)[\![F,F]\!]F^{n-2}.$$

Proof. We use induction and the previous lemma. For $n = 2$ the formula clearly holds by equation (2.1). Suppose that it holds for some $n \in \mathbb{N}_{>2}$, then

$$\begin{aligned}\Delta(F^{n+1}) &= \Delta(F \cdot F^n) = (\Delta F)F^n + [\![F,F^n]\!] + F\Delta(F^n)\\ &= (\Delta F)F^n + n[\![F,F]\!]F^{n-1} + nF(\Delta F)F^{n-1} + \frac{1}{2}n(n-1)F[\![F,F]\!]F^{n-2}\\ &= (n+1)(\Delta F)F^n + \frac{1}{2}(n+1)n[\![F,F]\!]F^{n-1},\end{aligned}$$

so that the statement also holds for $n+1$. □

Proof of Proposition 2.4. For convenience, we set $F = \frac{i}{\hbar}S^\hbar_{\mathrm{BV}}$. Then

$$\begin{aligned}0 = \Delta(\exp F) &= \Delta\left(\sum_{n=0}^\infty \frac{1}{n!}F^n\right) = \sum_{n=0}^\infty \frac{1}{n!}\Delta(F^n)\\ &= \sum_{n=0}^\infty \frac{n}{n!}(\Delta F)F^{n-1} + \sum_{n=0}^\infty \frac{1}{2n!}n(n-1)[\![F,F]\!]F^{n-2}\\ &= (\Delta F)\sum_{n=1}^\infty \frac{1}{(n-1)!}F^{n-1} + \frac{1}{2}[\![F,F]\!]\sum_{n=2}^\infty \frac{1}{(n-2)!}F^{n-2}\\ &= \left(\Delta F + \frac{1}{2}[\![F,F]\!]\right)\exp F = \left(\frac{i}{\hbar}\Delta S^\hbar_{\mathrm{BV}} - \frac{1}{2\hbar^2}[\![S^\hbar_{\mathrm{BV}}, S^\hbar_{\mathrm{BV}}]\!]\right)\exp\left(\frac{i}{\hbar}S^\hbar_{\mathrm{BV}}\right),\end{aligned}$$

from which the result follows. □

Proposition 2.7. *Take two elements $\mathcal{O}, S^\hbar_{\mathrm{BV}} \in A^\hbar$, and suppose that $S^\hbar_{\mathrm{BV}}$ is even. If $\mathcal{O}$ is a quantum observable and $S^\hbar_{\mathrm{BV}}$ satisfies $\Delta(\exp(iS^\hbar_{\mathrm{BV}}/\hbar)) = 0$, then $\mathcal{O}$ satisfies*

$$\Omega(\mathcal{O}) := [\![S^\hbar_{\mathrm{BV}}, \mathcal{O}]\!] - i\hbar\Delta\mathcal{O} = 0. \tag{2.7}$$

The operator Ω is known as the *quantum BV-differential*.

Proof. For convenience, let us set $F = \frac{i}{\hbar}S^\hbar_{\mathrm{BV}}$ again. We first calculate, using Lemma 2.5,

$$[\![\mathcal{O}, \exp F]\!] = \sum_{n=0}^\infty \frac{1}{n!}[\![\mathcal{O}, F^n]\!] = \sum_{n=0}^\infty \frac{n}{n!}[\![\mathcal{O}, F]\!]F^{n-1} = [\![\mathcal{O}, F]\!]\exp F.$$

Now $\Delta(\mathcal{O}\exp F) = 0$ since $\mathcal{O}$ is a quantum observable, so

$$\begin{aligned} 0 = \Delta(\mathcal{O}\exp F) &= (\Delta\mathcal{O})\exp F + [\![\mathcal{O}, \exp F]\!] + \mathcal{O}\Delta(\exp F) \\ &= \big((\Delta\mathcal{O}) + [\![\mathcal{O}, F]\!]\big)\exp F = \Big((\Delta\mathcal{O}) + \frac{i}{\hbar}[\![\mathcal{O}, S^\hbar_{\mathrm{BV}}]\!]\Big)\exp\Big(\frac{i}{\hbar}S^\hbar_{\mathrm{BV}}\Big), \end{aligned}$$

from which the assertion follows. □

Proposition 2.8. *Let $\mathcal{O} \in A^\hbar$, and let the even $S^\hbar_{\mathrm{BV}} \in A^\hbar$ satisfy the quantum master equation (2.6) for the BV-Laplacian Δ. Then the quantum BV-differential Ω, defined in (2.7), squares to zero*:

$$\Omega^2(\mathcal{O}) = 0.$$

Proof. We calculate

$$\begin{aligned} \Omega^2(\mathcal{O}) &= [\![S^\hbar_{\mathrm{BV}}, [\![S^\hbar_{\mathrm{BV}}, \mathcal{O}]\!] - i\hbar\Delta\mathcal{O}]\!] - i\hbar\Delta\big([\![S^\hbar_{\mathrm{BV}}, \mathcal{O}]\!] - i\hbar\Delta\mathcal{O}\big) \\ &= [\![S^\hbar_{\mathrm{BV}}, [\![S^\hbar_{\mathrm{BV}}, \mathcal{O}]\!]]\!] - i\hbar[\![S^\hbar_{\mathrm{BV}}, \Delta\mathcal{O}]\!] - i\hbar[\![\Delta S^\hbar_{\mathrm{BV}}, \mathcal{O}]\!] + i\hbar[\![S^\hbar_{\mathrm{BV}}, \Delta\mathcal{O}]\!] \\ &\quad + (i\hbar)^2\Delta^2\mathcal{O}. \end{aligned}$$

The last term vanishes identically since the Laplacian is a differential, while the second term cancels against the fourth term. Using the Jacobi identity for the bracket $[\![\,,\,]\!]$ on the first term, we obtain:

$$\Omega^2(\mathcal{O}) = \tfrac{1}{2}[\![[\![S^\hbar_{\mathrm{BV}}, S^\hbar_{\mathrm{BV}}]\!], \mathcal{O}]\!] - i\hbar[\![\Delta S^\hbar_{\mathrm{BV}}, \mathcal{O}]\!] = [\![\tfrac{1}{2}[\![S^\hbar_{\mathrm{BV}}, S^\hbar_{\mathrm{BV}}]\!] - i\hbar\Delta S^\hbar_{\mathrm{BV}}, \mathcal{O}]\!] = 0,$$

by the quantum master equation for $S^\hbar_{\mathrm{BV}}$. □

2.4 Products of integral functionals

From this section and onwards, we will once again work in the variational setup. Having generalized the Schouten bracket on a smooth manifold to the variational Schouten bracket in the spirit of secondary calculus, we note that the variational Schouten bracket seems to have lost a property that its smooth counterpart does satisfy: the fact that $[\![\,,\,]\!]$ is a derivation in each of its arguments (see Remark 1.11 on p. 15). In this section we shall investigate, and eventually restore this property for the variational Schouten bracket.

We start with a vector bundle $\pi\colon E \to M$ as usual. Then, as discussed previously, every equivalence class[7] $\mathcal{H} = \int h([q])\,\mathrm{d}^n x \in \overline{H}^n(\pi)$ determines a map $\mathcal{H}\colon \Gamma(\pi) \to \Bbbk$ (which we let be either $\mathbb{R}$ or, possibly, $\mathbb{C}$). Namely, for any $s \in \Gamma(\pi)$ we set

$$\mathcal{H}\colon s \mapsto \mathcal{H}(s) = \int_M j^\infty(s)^* h\big([q]\big)\,\mathrm{d}^n x = \int_M h(j^\infty_x(s))\,\mathrm{d}^n x \in \Bbbk. \tag{2.8}$$

[7] Here and onwards, if $h \in \mathcal{F}(\pi)$ is some function then we will sometimes write $h([q])$ to make explicit the dependence of h on the jet coordinates q^α and q^α_σ of $J^\infty(\pi)$.

Next, we introduce a multiplicative structure $F \otimes_{\Bbbk} G \mapsto F \cdot G$ by the rule

$$(F \cdot G)(s) := F(s)G(s), \quad s \in \Gamma(\pi),$$

for two integral functionals $F, G \in \overline{H}^n(\pi)$; here the product on the right hand side is that of $\Bbbk$. It is clear that under the viewpoint that sections are "points" this is the generalization to $\overline{H}^n(\pi)$ of the pointwise product on the algebra of smooth functions of a smooth manifold. There is, however, a problem: the space $\overline{H}^n(\pi)$ is not closed under this product, in the sense that generally $F \cdot G \notin \overline{H}^n(\pi)$. We solve this by taking (formal) sums of products of arbitrary finite numbers of integral functionals from $\overline{H}^n(\pi)$:

$$\mathfrak{M}(\pi) := \bigoplus_{i=1}^{\infty} \overline{H}^n(\pi)^{\otimes i} \subsetneq \mathrm{Map}\big(\Gamma(\pi) \to \Bbbk\big), \tag{2.9}$$

in the space $\mathrm{Map}(\Gamma(\pi) \to \Bbbk)$ of all maps which assign a number to every section.

Remark 2.9. By constructing the space of maps $\mathfrak{M}(\pi)$ in this fashion, we have not exited the realm of integral functionals on jet spaces. To see this, take two integral functionals $F = \int f(x, [q])\, \mathrm{d}^n x$ and $G = \int g(x', [q'])\, \mathrm{d}^n x'$ on π. Then we may write $F \cdot G$ as

$$F \cdot G = \int f(x, [q])\, g(x', [q'])\, \mathrm{d}^n x \wedge \mathrm{d}^n x'.$$

Now $f(x, [q])\, g(x', [q'])$ may be considered as a function on the jet space of $\pi \times \pi$, the product bundle of π with itself above the base manifold $M \times M$, of which the coordinates are x^i and x'^j respectively. The top cohomology of this space is then $\overline{H}^{2n}(\pi \times \pi)$, and we have $F \cdot G = \int fg\, \overline{\mathrm{d}}x \wedge \overline{\mathrm{d}}x' \in \overline{H}^{2n}(\pi \times \pi)$. Thus, $F \cdot G$ is an integral functional with respect to the bundle $\pi \times \pi$, and we see that the second term $\overline{H}^n(\pi) \otimes_{\Bbbk} \overline{H}^n(\pi)$ in the direct sum in (2.9) is a subspace of $\overline{H}^{2n}(\pi \times \pi)$. Continuing this line of reasoning to products of any number of factors we find

$$\overline{H}^n(\pi)^{\otimes i} \subset \overline{H}^{i \cdot n}(\underbrace{\pi \times \cdots \times \pi}_{i \text{ copies}}),$$

and therefore also

$$\mathfrak{M}(\pi) \subset \bigoplus_{i=1}^{+\infty} \overline{H}^{i \cdot n}(\underbrace{\pi \times \cdots \times \pi}_{i \text{ copies}}).$$

(Note that this is a strict inclusion; for example, $\overline{H}^{2n}(\pi \times \pi)$ contains functionals such as $\int q_{xx'}\, \mathrm{d}^n x \wedge \mathrm{d}^n x'$ while $\overline{H}^n(\pi) \otimes_{\Bbbk} \overline{H}^n(\pi)$ does not.) Thus, although composite but homogeneous functionals from $\mathfrak{M}(\pi)$ (i.e., functionals of the form $F = F_1 \cdots F_i$ for some i and $F_k \in \overline{H}^n(\pi)$ for $1 \leq k \leq i$) are not integral with respect to the bundle π, they are

integral with respect to another bundle, namely $\underbrace{\pi \times \cdots \times \pi}_{i \text{ copies}}$.

One of the major differences between ordinary derivatives $\partial/\partial x^i$ on $C^\infty(M)$ and the variational derivative $\delta/\delta q^\alpha$ on $\mathcal{F}(\pi)$ is that the former satisfies a Leibniz rule, but the latter does not. Equivalently, the de Rham differential d_{dR} satisfies the Leibniz rule but the Euler operator δ does not. The above interpretation of the "pointwise" product $F \cdot G$ of two functionals F, G allows us to formulate a variational analog of this Leibniz rule (that simultaneously mirrors the Leibniz rule that the *functional* derivative satisfies when acting on the product $F[s]G[s]$ of two functionals).

Proposition 2.10. *If F and G are two functionals on π then*

$$\delta(FG) = (\delta F)G + F(\delta G) \in E_1^{2n,1}(\pi \times \pi) \tag{2.10}$$

where the first δ is the Euler operator on $\pi \times \pi$, and the second and third is that of π.

Proof. Let us for ease of notation assume that the base space and fiber of π are both one-dimensional; the general result follows immediately by carefully restoring the indices in the proof below. Recall that the Euler operator is defined as $\delta\colon \omega \mapsto \int \mathrm{d}_{\mathcal{C}}\omega$. On $\pi \times \pi$, there are the base coordinates x and x' and fiber coordinates q and q'. Moreover, in the jet space $J^\infty(\pi \times \pi)$ the derivatives may mix; i.e., there are coordinates of the form $q_{xx'}$. Thus, the Cartan differential on $\pi \times \pi$ becomes

$$\mathrm{d}_{\mathcal{C}} = \mathrm{d}_{\mathcal{C}} q_\sigma \frac{\partial}{\partial q_\sigma} + \mathrm{d}_{\mathcal{C}} q'_\tau \frac{\partial}{\partial q'_\tau}$$

where both the multi-indexes σ and τ may contain both x and x'. We obtain

$$\begin{aligned}\delta(FG) &= \int \mathrm{d}_{\mathcal{C}}(fg)\, \overline{\mathrm{d}}x \wedge \overline{\mathrm{d}}x' \\ &= \int \Big(\frac{\partial(fg)}{\partial q_\sigma} \mathrm{d}_{\mathcal{C}} q_\sigma + \frac{\partial(fg)}{\partial q'_\tau} \mathrm{d}_{\mathcal{C}} q'_\tau \Big)\, \overline{\mathrm{d}}x \wedge \overline{\mathrm{d}}x'.\end{aligned}$$

Now f depends only on q and x, and g depends only on q' and x'. Therefore, although in the manifold $J^\infty(\pi \times \pi)$ the coordinates of the kind $q_{xx'}$ occur, the product fg does not in fact depend on them. Considering the term containing σ above, the consequence is that this term will be nonzero only if σ does not contain any x'. Of course, a similar statement holds for the term containing τ. Thus we get

$$\begin{aligned}\delta(FG) &= \int \Big(\frac{\partial f}{\partial q_\sigma} g\, \mathrm{d}_{\mathcal{C}} q_\sigma + f \frac{\partial g}{\partial q'_\tau}\, \mathrm{d}_{\mathcal{C}} q'_\tau \Big)\, \overline{\mathrm{d}}x \wedge \overline{\mathrm{d}}x' \\ &\cong \int \Big(\frac{\delta f}{\delta q} g\, \mathrm{d}_{\mathcal{C}} q + f \frac{\delta g}{\delta q'}\, \mathrm{d}_{\mathcal{C}} q' \Big)\, \overline{\mathrm{d}}x \wedge \overline{\mathrm{d}}x'.\end{aligned}$$

In the partial integration in the previous step, in the first term no derivatives fall on g, since σ contains only x because f only depends on x, while g only depends on x'. A

similar remark explains the second term. Lastly, as a notational trick we separate the integrals again, obtaining

$$\begin{aligned}\delta(FG) &= \int \frac{\delta f}{\delta q} \mathrm{d}_{\mathcal{C}} q \, \overline{\mathrm{d}}x \int g \, \overline{\mathrm{d}}x' + \int f \, \overline{\mathrm{d}}x \int \frac{\delta g}{\delta q'} \mathrm{d}_{\mathcal{C}} q' \, \overline{\mathrm{d}}x' \\ &= (\delta F)G + F(\delta G).\end{aligned}$$

□

In section 2.2 we compiled a list of variational generalizations of concepts on smooth manifolds; there we listed $\overline{H}^n(\pi)$ as the generalization of the ring of functions $C^\infty(M)$ on a manifold M. The latter being an algebra, it is now clear that the product $\cdot$ on $\mathfrak{M}(\pi)$ is the variational generalization of the product of $C^\infty(M)$.

Going further, let us now consider the Schouten bracket. To the initial bundle π we adjoin its parity-odd dual $\Pi\widehat{\pi}$, whose fiber coordinates we shall denote in this chapter not with b but with $q^\dagger$. Also anticipating future notations, we set $\pi_{\mathrm{BV}} = \pi \times \Pi\widehat{\pi}$, so that the arguments for the Schouten bracket are elements of $\overline{H}^n(\pi_{\mathrm{BV}})$. Moving now to the space $\mathfrak{M}(\pi_{\mathrm{BV}})$, we see that a natural variational generalization of the wedge product of multivectors on manifolds is the product $\cdot$ on $\mathfrak{M}(\pi_{\mathrm{BV}})$, solving the problem that was noted in Remark 1.11 on p. 15. For this reason, and also anticipating physical applications, we thus define the following, creating the variational generalization of equation (1.5).

Definition 2.11. Let $F, G, H \in \mathfrak{M}(\pi_{\mathrm{BV}})$ be (products of) variational multivectors on π. We inductively extend the Schouten bracket on $\overline{H}^n(\pi_{\mathrm{BV}})$ to the space $\mathfrak{M}(\pi_{\mathrm{BV}})$ by the following formula:

$$[\![F, G \cdot H]\!] = [\![F, G]\!] \cdot H + (-)^{(|F|-1)|G|} G \cdot [\![F, H]\!].$$

It is easy to see that this bracket on $\mathfrak{M}(\pi_{\mathrm{BV}})$ is still graded skew-symmetric and still satisfies the graded Jacobi identity.

2.5 Euler-Lagrange equations with gauge symmetries

The remainder of this chapter is exclusively concerned with Euler-Lagrange equations that have gauge symmetries. Therefore, in this section we examine the relevant concepts and the geometrical setup.

Take a bundle $\pi\colon E \to M$. Deviating from previous conventions, we temporarily denote the fiber coordinates not with q^α but with ϕ^a (we will need the letter q for the full collection of BV-coordinates later on). Fixing coordinates in the base space and fiber, a system of differential equation is then given by $\mathcal{E} = \{F_i([\phi]) = 0\}$, $i = 1, \dots, k$, where $F_i \in \mathcal{F}(\pi)$. Thus, the fiber coordinates ϕ^a here play the role of unknowns. The functions F_i together form a tuple $F([\phi]) = \big(F_1([\phi]), \dots, F_k([\phi])\big)$, which is then an element of a certain vector bundle of rank k over $J^\infty(\pi)$. In our case this bundle is always of the form $\pi_\infty^*(\xi_0)$ for a certain vector bundle $\xi_0\colon E_0 \to M$. Setting $P_0 := \Gamma(\pi_\infty^*(\xi_0))$, we have $F \in P_0$.

Now it may happen that the equations $F_i([\phi])$ are not independent, but themselves

satisfy some differential equation – independent of whether $[\phi]$ satisfies the equations $F_i([\phi]) = 0$ or not. Recalling that $F \in P_0$ is a section, we thus would have

$$\Phi(F) = 0 \in P_1,$$

where $P_1 = \Gamma(\pi_\infty^*(\xi_1))$ is another space of sections, also of a pullback of a bundle $\xi_1 \colon E_1 \to M$ to $J^\infty(\pi)$, and where $\Phi \colon P_0 \to P_1$ is a differential operator in total derivatives. If Φ is $\mathbb{R}$-linear, then it induces a map $\Phi \colon \overline{J^\infty_\pi}(\xi_0) = J^\infty(\pi) \times_M J^\infty(\xi_0) \to \pi_\infty^*(E_1)$, that we shall denote with the same symbol, by $\Phi([F], [\phi]) = \Phi(F)([\phi])$.

We consider the following example to see this machinery in action.

Example 2.12 (Electromagnetism). In the case of electromagnetism the independent variable is the gauge potential $A = A_i \,\mathrm{d}x^i$, which is a one-form on the (four-dimensional) Riemannian[8] manifold M. Defining the field strength $\mathcal{F} = \mathrm{d}A$, the equation of motion is then

$$\begin{aligned} F &= \mathrm{d} \star \mathcal{F} = \mathrm{d} \star \mathrm{d}A = 0 \in \Lambda^3(M), \\ \partial_i \mathcal{F}^{ij} &= 0, \end{aligned}$$

here the lower line is the upper line in coordinates; $\star$ is the Hodge star; and coordinates are raised using the Riemannian metric on M. The equations above tautologically satisfy

$$\begin{aligned} 0 &= \mathrm{d}F = \mathrm{d}^2 \star \mathcal{F} = \mathrm{d}^2 \star \mathrm{d}A \in \Lambda^4(M), \\ 0 &= \partial_j \partial_i \mathcal{F}^{ij} = 0, \end{aligned}$$

independent of whether A satisfies the equation of motion (the lower line, which again is the upper line in coordinates, always holds because $\partial_j \partial_i$ is symmetric in an exchange of i and j, while the two-form $\mathcal{F}^{ij}$ is antisymmetric in i and j). We now translate the above system to the formalism of jet space:

- The vector bundle containing the unknowns is $\pi \colon \Lambda^1(M) \to M$. Thus we take the infinite jet bundle $J^\infty(\pi)$ with respect to this bundle, with elements of the form $A = (A_i, A_{i;x}, \dots, A_{i;\sigma}, \dots) \in J^\infty(\pi)$. If A is such a jet, then as an abuse of notation we shall still write $A = A_i \,\mathrm{d}x^i \in \overline{\Lambda}^1(\pi)$ in the remainder of this example.
- F is a map that takes a jet A and maps it to $A \mapsto F(A) = \overline{\mathrm{d}} \star \overline{\mathrm{d}} A \in \overline{\Lambda}^3(\pi) = \pi_\infty^*(\Lambda^3(M) \to M))$. Thus we find $F \in P_0 = \Gamma\big(\pi_\infty^*(\Lambda^3(M) \to M)\big)$.
- Setting $P_1 = \Gamma\big(\pi_\infty^*(\Lambda^4(M) \to M)\big)$, we find that the relation between the equations of motion is encoded by

  $$P_1 \ni \Phi(F) = \overline{\mathrm{d}} F = \overline{\mathrm{d}}^2 \star \overline{\mathrm{d}} A;$$

  we see that $\Phi \colon P_0 \to P_1$ is an differential operator in total derivatives.

[8] In physics literature the manifold M is generally not Riemannian but pseudo-Riemannian, having a metric with signature $(-,+,+,+)$; this may be obtained by performing a Wick rotation in the time direction. For convenience we take M to be Riemannian in this example.

We shall return to this case in Example 2.18 on p. 41 (although there we consider the larger class of Yang-Mills equations, from which electromagnetism can be obtained by taking $G = U(1)$ as the structure group).

Setting $\Phi_1 = \Phi$, it may happen that there are relations $\Phi_2(F, \Phi_1) = 0$ between the relations Φ_1 and the equation F. Continuing, we may find relation $\Phi_i \in P_i$ between the lesser generations of relations $F, \Phi_1, \dots, \Phi_{i-1}$, resulting in a tower of spaces of sections P_i that become progressively larger. We will assume, however, that this tower is finite; there are only $\lambda \in \mathbb{N}$ relations.

Now we specialize to *Euler-Lagrange equations*. A partial differential equation $\mathcal{E} = \{F([\phi]) = 0\}$ is Euler-Lagrange if it is of the form

$$\mathcal{E} = \{\delta S = 0 \in \widehat{\varkappa}(\pi)\}$$

for a certain $S = \int L \, \mathrm{d}^n x \in \overline{H}^n(\pi)$, which is called the *action*; the function L (or sometimes the density $L \, \mathrm{d}^n x$) is called its *Lagrangian*. The $\mathcal{F}(\pi)$-module containing the equation P_0 is the range of the Euler operator; that is, $P_0 = \widehat{\varkappa}(\pi)$. Suppose that there is a relation $\Phi(F) = 0 \in P_1$ between the equation $F = \delta S$, and suppose moreover that this relation is linear (which indeed seems to be the case in the majority of the physics literature). Recall that (see section 1.1.6 on p. 9) the adjoint of P_1 is defined as $\widehat{P}_1 = \mathrm{Hom}_{\mathcal{F}(\pi)}(P_1, \overline{\Lambda}^n(\pi))$; we denote the coupling between $\widehat{P}_1$ and P_1 by $\langle \, , \rangle$ as usual. Having defined all the notations, we can now formulate and prove a particularly elegant version of Noether's second theorem.

Theorem 2.13 (Noether's second theorem). *Consider the adjoint map* $\Phi^\dagger \colon \widehat{P}_1 \to \widehat{P}_0 = \widehat{\widehat{\varkappa}(\pi)} = \varkappa(\pi)$ *of the linear operator in total derivatives* $\Phi \colon P_0 \to P_1$. *Then* $\Phi^\dagger(\epsilon)$ *is a Noether symmetry of the action* S *for any* $\epsilon \in \widehat{P}_1$, *and therefore a symmetry of the Euler-Lagrange equation* $\delta S = 0$.

Proof. Taking any element $\epsilon \in \widehat{P}_1$, we trivially have

$$0 = \int \langle \epsilon, \Phi(F) \rangle \cong \int \langle \Phi^\dagger(\epsilon), F \rangle = \int \langle \Phi^\dagger(\epsilon), \delta S \rangle \cong \partial^{(\phi)}_{\Phi^\dagger(\epsilon)}(S);$$

although these are actually all equalities, we use the symbol $\cong$ to indicate that an integration by parts took place under the integration sign. The equation above says that $\Phi^\dagger(\epsilon) \in \varkappa(\pi)$ is a Noether symmetry of the equation for any $\epsilon \in \widehat{P}_1$. The second claim holds because any Noether symmetry of the action S induces a symmetry of the equation $\delta S = 0$. □

We should remark that Φ need not be linear in order for Noether's second theorem to hold. For a proof of the general nonlinear case, see e.g., [Kis12c, Ch. 6].

Let us set $\pi_0 = \pi$ and $\pi_i = \xi_i$. Proceeding in building the setup for the BV-Laplacian, we now take the parity-reversed covectors associated to the vectors[9] $\partial^{(\phi)}_\varphi$, as before in

[9] Note that the *vector* bundle π is used to introduce the physical fields $\phi^1, \dots, \phi^m$; however, these objects could be, for example, components of a *covector* $A = \sum_{\alpha=1}^n \phi^\alpha \, \mathrm{d}x^\alpha$ of a gauge connection's one-form in a fiber bundle over M with a given structure Lie group (see Examples 2.12 and 2.18 on pages 37 and 41, respectively).

Section 1.3, yielding the horizontal jet bundle $\overline{J^\infty_{\pi_0}}(\Pi\widehat{\pi}_0) = J^\infty(\pi_0 \times_M \Pi\widehat{\pi}_0)$. We will denote the new odd fiber coordinates with $\phi^\dagger_a$ and $\phi^\dagger_{a,\sigma}$ for their derivatives; these are called the *antifields*. Passing to the bundle $P_1 = \Gamma(\overline{J^\infty_{\pi_0}}(\pi_1))$ containing the Noether identities, we denote the fiber coordinates in π_1 with $\gamma^{\dagger,\mu}$ (the *antighosts*). Adjoining the parity-reversed dual bundle $\Pi\widehat{P}_1$ as well, whose fiber coordinates are the odd *ghosts* γ_μ, we obtain the jet space $J^\infty(\pi_0 \times_M \Pi\widehat{\pi}_0 \times_M \pi_1 \times_M \Pi\widehat{\pi}_1)$.

Continuing this reasoning, the higher order Noether relations between the other relations give rise to $\lambda + 1$ sets of pairs of dual, opposite-parity bundles $\pi_i \leftrightarrow \Pi\widehat{\pi}_i$. Denoting the Whitney sums

$$\boldsymbol{\pi} = \pi_0 \times_M \cdots \times_M \pi_\lambda \qquad \text{and} \qquad \pi_{\mathrm{BV}} = \boldsymbol{\pi} \times_M \Pi\widehat{\boldsymbol{\pi}},$$

we thus have constructed the Batalin–Vilkovisky (BV) superbundle

$$(\pi_{\mathrm{BV}})_\infty \colon J^\infty(\boldsymbol{\pi}) \times_M J^\infty(\Pi\widehat{\boldsymbol{\pi}}) \longrightarrow M.$$

We denote by

$$q^\alpha = (\phi^a, \gamma^{\dagger,\mu}, c^\dagger_\nu, \dots)$$

the entire set of even-parity BV-coordinates and by

$$q^\dagger_\alpha = (\phi^\dagger_a, \gamma_\mu, c^\nu, \dots)$$

their odd-parity duals.

Remark 2.14. In the terminology of the previous chapter, elements $\mathcal{H} \in \overline{H}^n(\pi_{\mathrm{BV}})$ would be multivectors of the bundle $\boldsymbol{\pi}$, either nonhomogeneous or having a certain degree $|\mathcal{H}|$. Here and onwards we will just call them integral functionals. This notion of the degree of functionals coincides with the *ghost numbers* or *ghost parity* from the physics literature; there the degree of $\mathcal{H}$ is often written as $\epsilon(\mathcal{H})$ or $\mathrm{gh}(\mathcal{H})$.

The entire collection of auxiliary bundles – that is, π_0 and π_i for all $i = 1, \dots, \lambda$ – is even with respect to the ghost parity. The ghost parities of objects that belong to P_i and $\Pi\widehat{P}_i$ at each i are always opposite: the former are ghost-parity even and the latter are odd. Should one consider an independently graded setup of fields ϕ^a as fibers of a superbundle $\pi = (\pi^{\bar{0}}|\pi^{\bar{1}})$ or a setup with $\mathbb{Z}$-graded differential forms taken as fields (see [CF00]), and should one then introduce the action functional $S \in \overline{H}^n(\pi)$, a count of extra signs in the equations of motion, Noether identities etc., would be tedious but straightforward (c.f. Example 2.19 on page 41).

2.6 A Laplacian

In this final section we will examine a candidate BV-Laplacian Δ on $\overline{H}^n(\pi_{\mathrm{BV}})$. Contrary to the conventional BV-Laplacian from the physical literature it will involve no delta-functionals or infinities, but unfortunately, it will turn out that with respect to the

variational Schouten bracket on $\mathfrak{M}(\pi_{\mathrm{BV}})$, it is not a Laplacian in the sense of Definition 2.1 on p. 28.

Definition 2.15. Let $\mathcal{H} = \int h \, \mathrm{d}^n x \in \overline{H}^n(\pi_{\mathrm{BV}})$ be an integral functional, possibly depending on the entire collection of BV-variables and on their derivatives up to arbitrarily high order. We define a Laplacian $\Delta\colon \overline{H}^n(\pi_{\mathrm{BV}}) \to \overline{H}^n(\pi_{\mathrm{BV}})$ on the space of integral functionals as follows:

$$\Delta(\mathcal{H}) = \int \frac{\overrightarrow{\delta}}{\delta q^\alpha} \left(\frac{\overrightarrow{\delta h}}{\delta q^\dagger_\alpha} \right) \mathrm{d}^n x. \tag{2.11}$$

Remark 2.16. Consider the right variational derivative in the defining equation (2.11) for the Laplacian:

$$\frac{\overrightarrow{\delta}}{\delta q^\dagger_\alpha} = \frac{\overrightarrow{\partial}}{\partial q^\dagger_\alpha} + \sum_{\sigma \neq \varnothing} (-)^\sigma D_\sigma \circ \frac{\overrightarrow{\partial}}{\partial q^\dagger_{\alpha,\sigma}}$$

To the immediate left of this in equation (2.11) stands the next variational derivative, $\overrightarrow{\delta}/\delta q^\alpha$. Since $\overrightarrow{\delta}/\delta q^\alpha \circ D_\sigma = 0$ for any α and σ, all of the terms in the sum over $\sigma \neq \varnothing$ above disappear, and the right variational derivative $\overrightarrow{\delta}/\delta q^\alpha$ becomes just $\overrightarrow{\partial}/\partial q^\dagger_\alpha$ in equation (2.11). Moreover, all terms of the left variational derivative that contain total derivatives do not contribute to the functional, because the integral over a total derivative is zero according to our convention on no boundary terms. Therefore, equation (2.11) reduces to

$$\Delta(\mathcal{H}) = \int \frac{\overrightarrow{\partial}}{\partial q^\alpha} \frac{\overrightarrow{\partial h}}{\partial q^\dagger_\alpha} \mathrm{d}^n x. \tag{2.12}$$

Thus, we find that our Laplacian is insensitive to all jet coordinates with derivatives q^a_σ. This results in some serious problems that we will discuss in Counterexample 2.20 on p. 42. First, however, we note that this observation has the following consequence.

Proposition 2.17. *The linear operator $\Delta\colon \overline{H}^n(\pi_{\mathrm{BV}}) \to \overline{H}^n(\pi_{\mathrm{BV}})$ is a differential: $\Delta^2 = 0$.*

Proof. Using the observation above twice, we obtain

$$\Delta^2(\mathcal{H}) = \int \frac{\overrightarrow{\partial}}{\partial q^\alpha} \frac{\overrightarrow{\partial}}{\partial q^\dagger_\alpha} \frac{\overrightarrow{\partial}}{\partial q^\beta} \frac{\overrightarrow{\partial h}}{\partial q^\dagger_\beta} \mathrm{d}^n x.$$

Since the middle two partial derivatives in this expression commute, this becomes

$$\Delta^2(\mathcal{H}) = \int \frac{\overrightarrow{\partial}}{\partial q^\alpha} \frac{\overrightarrow{\partial}}{\partial q^\beta} \frac{\overrightarrow{\partial}}{\partial q^\dagger_\alpha} \frac{\overrightarrow{\partial}}{\partial q^\dagger_\beta} (h) \, \mathrm{d}^n x.$$

This is the composition of an expression which is symmetric in α and β (the left two partial derivatives) and an expression which is antisymmetric in α and β (the right two partial derivatives). Therefore it is zero. □

Although it will in general turn out to be a problem that the Laplacian Δ is insensitive to derivative coordinates q^α_σ (and $q^\dagger_{\alpha,\sigma}$), this is of course not an issue when the functional to which the Laplacian is applied does not contain any derivative coordinates – or when the terms that do contain them do not contribute, as is the case in the following example.

Example 2.18. Take a compact, semisimple Lie group G with Lie algebra $\mathfrak{g}$ and consider the corresponding Yang-Mills theory (c.f. Example 2.12 on p. 37, where $G = U(1)$). Write A^a_i for the (coordinate expression of) the gauge potential A – a lower index i because A is a one-form on the base manifold (i.e., a covector), and an upper index a because A is a vector in the Lie algebra $\mathfrak{g}$ of the Lie group G. Defining the field strength $\mathcal{F}$ by $\mathcal{F}^a_{ij} = \partial_i A^a_j - \partial_j A^a_i + f^a_{bc} A^b_i A^c_j$ where f^a_{bc} are the structure constants of $\mathfrak{g}$, the Yang-Mills action is

$$S_0 = \frac{1}{4} \int \mathcal{F}^a_{ij} \mathcal{F}^{a,ij} \, \mathrm{d}^n x.$$

Denoting the parity-odd ghosts from $\Pi\widehat{P}_1$ by γ^a and the even antighosts from P_1 by $\gamma^\dagger_c$, the full BV-action S_{BV} is

$$S_{\mathrm{YM}} = S_0 + \int A^{i\dagger}_a (D_i \gamma^a + f^a_{bc} A^b_i \gamma^c) \, \mathrm{d}^4 x - \frac{1}{2} \int f^c_{ab} \gamma^a \gamma^b \gamma^\dagger_c \, \mathrm{d}^4 x.$$

Let us calculate the Laplacian of this functional. As a consequence of equation (2.11), the only terms which survive in $\Delta(S_{\mathrm{YM}})$ are those which contain both A and $A^\dagger$, or both γ and $\gamma^\dagger$ (this is so for both our Laplacian Δ, and the BV-Laplacian from the literature). Therefore,

$$\begin{aligned}
\Delta(S_{\mathrm{YM}}) &= \int \left(\frac{\overrightarrow{\delta}}{\delta A^d_j} \frac{\overrightarrow{\delta}}{\delta A^{j\dagger}_d} (f^a_{bc} A^{i\dagger}_a A^b_i \gamma^c) - \frac{1}{2} \frac{\overrightarrow{\delta}}{\delta \gamma^\dagger_d} \frac{\overrightarrow{\delta}}{\delta \gamma^d} (f^c_{ab} \gamma^a \gamma^b \gamma^\dagger_c) \right) \mathrm{d}^4 x \\
&= \int \left(\frac{\overrightarrow{\delta}}{\delta A^d_j} (f^d_{bc} A^b_j \gamma^c) - \frac{1}{2} \frac{\overrightarrow{\delta}}{\delta \gamma^\dagger_d} (f^c_{db} \gamma^b \gamma^\dagger_c - f^c_{ad} \gamma^a \gamma^\dagger_c) \right) \mathrm{d}^4 x \\
&= \int \left(f^d_{dc} \gamma^c - \frac{1}{2} (f^d_{db} \gamma^b - f^d_{ad} \gamma^a) \right) \mathrm{d}^4 x = 0.
\end{aligned}$$

Example 2.19. Consider the nonlinear Poisson sigma model introduced in [CF00]. Since its fields are not all purely even, we would have to generalize all of our reasoning so far to a $\mathbb{Z}_2$-graded setup – which is, as noted before, tedious but straightforward. A calculation of $\Delta(S_{\mathrm{CF}})$ of the BV-action S_{CF} of this model would, up to minor differences in conventions and notations, proceed just as it does in the paper itself, in section 3.2 – except that no infinite constants or delta functions appear.

Contrary to the conventional BV-Laplacian, the Laplacian Δ defined here does not always

satisfy the equation $\Delta([\![F,G]\!]) = [\![\Delta F, G]\!] + (-)^{|F|-1}[\![F,\Delta G]\!]$. The reason is that if we were to proceed with the calculation of $\Delta([\![F,G]\!])$, it is not permissible to swap the symbols $\delta/\delta q^\alpha$ – which here stand for the *variational* derivative – with each other. To see this explicitly, take the following counterexample.

Counterexample 2.20. Let the base and fiber both be one-dimensional, and set

$$F = \int q^\dagger q q_{xx}\,\mathrm{d}x \quad \text{and} \quad G = \int q^\dagger_{xx}\cos q\,\mathrm{d}x.$$

Let f and g be the two integrands. We calculate

$$\frac{\overleftarrow{\delta f}}{\delta q} = q^\dagger q_{xx} + D_x^2(q^\dagger q) = q^\dagger q_{xx} + q^\dagger_{xx} q + 2q^\dagger_x q_x + q^\dagger q_{xx} = 2q^\dagger q_{xx} + q^\dagger_{xx} q + 2q^\dagger_x q_x, \tag{2.13}$$

$$\frac{\overleftarrow{\delta f}}{\delta q^\dagger} = q q_{xx}, \qquad \frac{\overrightarrow{\delta g}}{\delta q} = -q^\dagger_{xx}\sin q,$$

$$\frac{\overrightarrow{\delta g}}{\delta q^\dagger} = D_x^2(\cos q) = D_x(-q_x\sin q) = -q_{xx}\sin q - q_x^2\cos q.$$

(Note that since all four variational derivatives contain at most one parity-odd $q^\dagger$ or its derivatives, the directions of the arrows do not actually matter – i.e., switching their direction does not result in minus signs.) Consider $\Delta([\![F,G]\!])$. As was noted in Remark 2.16 on p. 40, we may write $\Delta(\mathcal{H}) = \int(\partial/\partial q)\circ(\partial/\partial q^\dagger)(\mathcal{H})$ for any $\mathcal{H}\in\overline{H}^n(\pi_{\mathrm{BV}})$; therefore, only terms in which $[\![F,G]\!]$ carries at least one $q^\dagger$ without derivatives with respect to the base space survive. This implies that in the first term of the bracket, $\overrightarrow{\delta f}/\delta q\cdot\overleftarrow{\delta g}/\delta q^\dagger$, the second and third term of the right hand side of (2.13) do not contribute, so we need not take them into account:

$$[\![F,G]\!] = \int\left(2q^\dagger q_{xx}(-q_{xx}\sin q - q_x^2\cos q) + \cdots - q q_{xx}\cdot(-q^\dagger_{xx}\sin q)\right)\mathrm{d}x,$$

where the dots indicate the omitted terms. For the same reason, the last term also does not contribute.

We calculate

$$\begin{aligned}\Delta([\![F,G]\!]) &= -2\int\frac{\partial}{\partial q}\frac{\partial}{\partial q^\dagger}(q^\dagger q_{xx}^2\sin q + q^\dagger q_{xx}q_x^2\cos q)\,\mathrm{d}x\\ &= -2\int\frac{\partial}{\partial q}(q_{xx}^2\sin q + q_{xx}q_x^2\cos q)\,\mathrm{d}x\\ &= -2\int(q_{xx}^2\cos q - q_{xx}q_x^2\sin q)\,\mathrm{d}x,\end{aligned}$$

whose integrand is not cohomogically trivial (as may be seen by calculating its variational derivative, which gives nonzero).

Now $\Delta F = \int q_{xx}\,\mathrm{d}x \cong 0$, so $[\![\Delta F, G]\!] = 0$. On the other hand, g has no q^+ without any x-derivatives in it so $\Delta G = 0$, so $[\![F, \Delta G]\!] = 0$ as well. In conclusion,

$$[\![\Delta F, G]\!] + (-)^{|F|-1}[\![F, \Delta G]\!] = 0 \neq \Delta([\![F, G]\!]).$$

Thus, we find that

$$\Delta([\![F, G]\!]) = [\![\Delta F, G]\!] + (-)^{|F|-1}[\![F, \Delta G]\!], \tag{2.14}$$

does not hold. Moreover, this implies that the triad $(\mathfrak{M}(\pi_{\mathrm{BV}}), [\![\,,]\!], \Delta)$ cannot be a BV-algebra in the sense of Definition 2.1 on p. 28. Indeed, suppose that $(\mathfrak{M}(\pi_{\mathrm{BV}}), [\![\,,]\!], \Delta)$ is a BV-algebra, then the equation above would be a consequence of

$$[\![F, G]\!] = (-)^{|F|}\left(\Delta(FG) - \Delta(F)G - (-)^{|F|}F\Delta G\right), \tag{2.15}$$

as noted in Proposition 2.2 on p. 28. Since (2.14) does not hold, we conclude that it is impossible for (2.15) to hold. Contrary to the case of the Schouten bracket, which we extended to products of functionals by a definition, we cannot solve the problem by simply defining $\Delta(FG)$ to be such that equation (2.15) becomes true.

Remark 2.21. Let us briefly attempt to give an intuitive, non-rigorous reason as to why our approach did not work. In Remark 2.9 we have seen that the product of two integral functionals $F \cdot G$ is not itself an integral functional, but an element of $\overline{H}^{2n}(\pi \times \pi)$. That is, each integral functional lives on its own separate copy of the bundle, and when we multiply them, these bundles remain separate (up until we evaluate it on the diagional $F[q]G[q]$).

We believe that the reason why a Laplacian defined in terms of a double functional works (at the cost of involving delta-functions and infinite constants), while one defined in terms of a double variational derivative does not work, is similar in nature.

Consider the defining formula for the functional derivative $\frac{\delta F[q]}{\delta q^\alpha(x)}$ of a functional:

$$\int \frac{\delta F[q]}{\delta q^\alpha(x)}\,\phi(x)\,\mathrm{d}x = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} F[q^\alpha + t\phi],$$

where the test function ϕ must have compact support but may otherwise be arbitrary. Taking a second functional derivative, we obtain

$$\iint \frac{\delta}{\delta q^\beta(y)}\frac{\delta F[q]}{\delta q^\alpha(x)}\,\phi(x)\psi(y)\,\mathrm{d}x\,\mathrm{d}y = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0}\left.\frac{\mathrm{d}}{\mathrm{d}\tilde{t}}\right|_{\tilde{t}=0} F[q + t\phi + \tilde{t}\psi].$$

The right hand side makes it immediately clear that, in contrast with variational derivatives, taking functional derivatives commute. From the left hand side we can see a different reason for this phenomenon: the two functional derivatives refer to two different integrals; i.e., two different geometries, as is the case for the product of two functionals as noted above. Because they are separate from each other in this way, we can freely swap them, just as one can swap partial derivatives of a smooth function.

Consider now a double variational derivative of some function. The two variational derivatives now refer to the same geometry, and are not insensitive to each other the way functional derivatives are. Indeed, we have seen in Remark 2.16 on p. 40 that the left variational derivative of the Laplacian defined in (2.11) on p. 40 eats the total derivatives coming from the right variational derivative, while the integration over the base space removes the total derivatives of the left variational derivative. From a more phenomenological perspective, we might say that what a double variational derivative calculates does not correspond to our intuition and expectancy of taking multiple derivatives.

As a result of this, our Laplacian is insensitive to any dependence of its argument on derivative coordinates, which is a very strange thing indeed to have in a variational setup. Indeed, since we have already seen that the Schouten bracket *is* sensitive to the presence of derivative coordinates, it is perhaps unreasonable to expect the Laplacian and Schouten bracket to combine to any reasonable structure such as a BV-algebra.

We suspect, then, that what we need is the following: a generalization or re-interpretation of the variational derivative that, like functional derivatives, brings with it its own geometry, so that when applied twice these geometries do not interfere with each other. Of course, such a double application of this operator should be sensitive to derivative coordinates. At the same time we do not want it to involve delta-functions when applied once, twice, or any number of times. We remain hopeful that such an operator may be defined in terms of jet bundles. Note that the Euler operator satisfies a Leibniz rule (see Proposition 2.10), while the variational derivative does not, so in that sense the Euler operator is more close to the functional derivative. Perhaps the operator that we are looking for, then, is the Euler operator or some close relative. Unfortunately, we did not have time to pursue this any further.

We will briefly return to these matters in Remark 4.5 on p. 74.

Chapter 3

Deformation quantization and the dual of Lie algebras

Deformation quantization is one of the ways in which one can convert a classical physics theory into a quantum mechanical theory. In this chapter, we review the technique of quantization deformation and its relation to the Hochschild complex of the ring of functions, after which we give a thorough explanation of the Kontsevich deformation quantization of the dual of Lie algebras.

3.1 Introduction

In physics, quantization refers to the process of taking a classical (that is, non-quantum-mechanical) theory of a physical system, and converting it into a quantum mechanical theory of the same system. There are many methods for achieving this, varying in complexity, how they work, and how rigorous they are. One technique, whose origin seems to be the influential papers [Bay+77; Bay+78a; Bay+78b], is called *deformation quantization*; it revolves around the idea that the quantum mechanical space of observables should be understood as a deformation of the pointwise product of the algebra of classical observables $C^\infty(M)$ on the manifold M. The deformation parameter is the Planck constant $\hbar$ (which here is just a formal parameter, instead of its physical value). Intuitively, the act of deformation quantization then is to "restore" the higher-order terms with respect to $\hbar$ that have been neglected in the classical version of the theory. In order to retain the correspondence with the original classical system, the deformation quantization will have to be compatible with the mathematical structures in terms of which the physics of the classical system was described.

For some types of manifolds, notably symplectic manifolds, it has been known for some time how to achieve deformation quantization. Then, in 1997, Maxim Kontsevich

posted an article to the arXiv (see http://arxiv.org/abs/q-alg/9709040) that was published in *Letters in Mathematical Physics* a full six years later [Kon03], in which he proved that any Poisson manifold can be deformed quantized. In this article Kontsevich briefly considers the resulting deformation quantization of the dual $\mathfrak{g}^*$ of a Lie algebra $\mathfrak{g}$, which has a natural Poisson structure induced by the Lie bracket on $\mathfrak{g}$. We will extensively review this case in section 3.6.

Before that, we first define star products in general in section 3.2, and then the Kontsevich star product in section 3.3. In section 3.4 we examine the automorphisms of star products, arriving at the concept of gauge transformations; after that we study Hochschild cohomology and its relation to star products in section 3.5.

Let us first give a brief description of the physical meaning of Poisson manifolds: the following is known as the Hamiltonian formalism. Let M be a Poisson manifold with Poisson-bivector $\alpha \in \bigwedge^2 TM$ so that $\{f,g\} = \alpha(\mathrm{d}f, \mathrm{d}g)$. If $H \in C^\infty(M)$ is a function, then $X_H := \{\cdot, H\} = \alpha(\mathrm{d}\cdot, \mathrm{d}H)$ is a vector field, called the *Hamiltonian vector field* of H. One is then interested in the integral curves $\gamma\colon I \to M$ along X_H; that is, solutions of the differential equation

$$\dot{\gamma}(t) = X_H(\gamma(t)). \tag{3.1}$$

Thus, the dynamics of the system are determined by the function H and the Poisson bracket $\{\,,\}$ on M.

Example 3.1. If H is the Hamiltonian of a classical system, and the coordinates on phase space $\mathbb{R}^2$ are p, q, then Hamilton's equations read

$$\dot{q} = \frac{\partial H}{\partial p}, \qquad \dot{p} = -\frac{\partial H}{\partial q}.$$

Setting $\gamma(t) = (p(t), q(t))$ then this is the same as

$$\begin{aligned}\dot{\gamma}(t) &= \dot{p}(t)\,\frac{\partial}{\partial p}\bigg|_{\gamma(t)} + \dot{q}(t)\,\frac{\partial}{\partial q}\bigg|_{\gamma(t)} \\ &= -\frac{\partial H}{\partial q}(t)\,\frac{\partial}{\partial p}\bigg|_{\gamma(t)} + \frac{\partial H}{\partial p}(t)\,\frac{\partial}{\partial q}\bigg|_{\gamma(t)} = X_H(\gamma(t))\end{aligned}$$

where X_H is the vector field $X_H = -\frac{\partial H}{\partial q}\frac{\partial}{\partial p} + \frac{\partial H}{\partial p}\frac{\partial}{\partial q}$. With respect to the Poisson structure $\alpha = -\frac{\partial}{\partial p} \wedge \frac{\partial}{\partial q}$ (which is dual to the symplectic two-form $\omega = \mathrm{d}p \wedge \mathrm{d}q$) we precisely have $X_H = \{\cdot, H\}$.

A deformation quantization of such a Hamiltonian system would be an associative *star product*

$$(f,g) \mapsto f \star g$$

on the ring $C^\infty(M)[\![\hbar]\!] \ni f, g$ of formal power series over $C^\infty(M)$; the star product $\star$ has to be such that the zeroth-order term of $f \star g$ is the pointwise product fg while the

first-order term is the Poisson bracket $\{f, g\}$. We will describe this in more detail in the next section, after the following example.

Example 3.2. For a first example of a star product, we begin with $V := \mathbb{R}^3$, together with the cross product $(v, w) \mapsto v \times w \in V$ of two vectors. This antisymmetric product features prominently in areas of physics such as mechanics and electromagnetism, and it has many generalizations to more abstract spaces. It is, however, not associative (instead, it satisfies the Jacobi identity). One of the generalizations of this cross product is a family of related products, called *star products*. Although these products are generally not commutative or anticommutative, they are, contrary to the cross product, always associative.

We proceed as follows. First, we take the *symmetric algebra* $S(\mathbb{R}^3)$ of $\mathbb{R}^3$, whose product (that we denote by $\cdot$ or contatenation) is commutative and associative but satisfies no other relations. We extend the cross product $\times$ of $\mathbb{R}^3$ to $S(\mathbb{R}^3)$ inductively by the following equation (the *Leibniz rule*):

$$(uv) \times w = u(v \times w) + v(u \times w).$$

(the brackets indicate which product is to be performed first). Requiring that $\times$ stays anticommutative on $S(\mathbb{R}^3)$ fixes it completely.

Writing v^n for the symmetric product of n vectors v, and $v^{\times n}$ for the cross product of n vectors v, we have the following.

Theorem 3.3. *For each $\hbar \in \mathbb{R}$ there is an associative* star product $\star$ *on $S(\mathbb{R}^3)$, satisfying for any $v, w \in \mathbb{R}^3$:*

$$v^n \star w = \sum_{k=0}^{n} \hbar^k \binom{n}{k} \widehat{B}_k \, v^{n-k}(v^{\times k} \times w),$$

where the $\widehat{B}_k$ are the (first) Bernoulli numbers.[1] *In particular,*

$$v \star w = vw + \tfrac{\hbar}{2}(v \times w).$$

Thus, for example,

$$v^2 \star w = v^2 w + \hbar v(v \times w) + \tfrac{\hbar^2}{6} v \times v \times w. \tag{3.2}$$

The two equations in the theorem above in fact determine the star product $x \star y$ for all $x, y \in S(\mathbb{R}^3)$.

This is an example of the Kontsevich product on Lie algebras.[2] We shall discuss these

[1] The *(first) Bernoulli numbers* are the coefficients of the Taylor series of $-x/(e^{-x} - 1)$:

$$\frac{-x}{e^{-x} - 1} = \sum_{k=0}^{\infty} \frac{x^k}{k!} \widehat{B}_k.$$

The first four of these numbers are $B_0 = 1$, $B_1 = 1/2$, $B_2 = 1/6$, $B_4 = -1/30$, while $B_k = 0$ for any odd k larger than 1.

[2] Actually, a star product which is gauge equivalent to the Kontsevich product.

in more general form, and prove the theorem above, in section 3.6.3 on p. 65, and we will return to this example in Example 3.34 on p. 68.

3.2 Star products

Let M be a smooth manifold; we will denote with $A = C^\infty(M)$ its ring of smooth functions, which is an associative algebra over $\mathbb{R}$. We consider the space of formal power series $A[\![\hbar]\!]$ over A in the parameter $\hbar$.

Definition 3.4. A *star product* is an $\mathbb{R}[\![\hbar]\!]$-linear product on $A[\![\hbar]\!]$, which for $f, g \in A$ takes the following form:

$$(f,g) \mapsto f \star g = fg + \hbar B_1(f,g) + \hbar^2 B_2(f,g) + \cdots = \sum_{i=0}^{\infty} \hbar^i B_i(f,g),$$

where $B_0(f,g) = fg$, such that

- the star product is associative,
- the constant function $1 \in A$ is a unit: $1 \star f = f \star 1 = f$ for all $f \in A$,
- B_i is for each i a bilinear differential operator of bounded order, i.e. they are of the form

  $$B_i(f,g) = \sum_{K,L} \beta_i^{KL} (\partial_K f)(\partial_L g).$$

  Here K, L are multi-indices of any length; ∂_K is the usual notation for higher order derivatives; and β_i^{KL} is smooth for all its indices, and nonzero for only finitely many choices of K, L and i.

How the star product acts on arbitrary elements of $A[\![\hbar]\!]$ (that is, formal power series of smooth functions) follows from the condition of $\mathbb{R}[\![\hbar]\!]$-linearity:

$$\begin{aligned}\left(\sum_{n\geq 0} \hbar^n f_n\right) \star \left(\sum_{m\geq 0} \hbar^m g_m\right) &= \sum_{n,m} \hbar^{n+m} f_n \star g_m \\ &= \sum_{n,m} \hbar^{n+m} f_n g_m + \sum_{n,m,l} \hbar^{n+m+l} B_l(f_n, g_m).\end{aligned}$$

There are some immediate consequences to this definition. First, from $f = 1 \star f = 1 \cdot f + \sum_{n\geq 1} \hbar^n B_n(1,f)$ it follows that, for any $n \geq 1$,

$$B_n(1,f) = B_n(f,1) = 0,$$

and by linearity it follows that B_n always gives zero whenever one of its arguments is constant. Moreover, we have the following.

Proposition 3.5. *B_1 satisfies for all $f, g, h \in A$*

$$fB_1(g,h) - B_1(fg,h) + B_1(f,gh) - B_1(f,g)h = 0. \tag{3.3}$$

This means precisely that the map $B_1 \colon A \otimes A \to A$ is a 2-cocycle in the Hochschild complex of A, which will be discussed later (see Section 3.5 starting on p. 56).

Proof. Working out associativity up to and including first order, we have

$$\begin{aligned}(f \star g) \star h &= (fg + \hbar B_1(f,g) + O(\hbar^2)) \star h \\ &= fgh + \hbar B_1(f,g)h + \hbar B_1(fg,h) + O(\hbar^2), \\ f \star (g \star h) &= f \star (gh + \hbar B_1(g,h) + O(\hbar^2)) \\ &= fgh + \hbar f B_1(g,h) + \hbar B_1(f,gh) + O(\hbar^2).\end{aligned}$$

Subtracting the upper equation from the lower one gives

$$0 = \hbar(fB_1(g,h) + B_1(f,gh) - B_1(fg,h) - B_1(f,g)h) + O(\hbar^2),$$

from which the result follows. □

Let us decompose B_1 into an symmetric and antisymmetric part: $B_1^+(f,g) = \frac{1}{2}(B_1(f,g) + B_1(g,f))$ and $B_1^- = \frac{1}{2}(B_1(f,g) - B_1(g,f))$.

Proposition 3.6. *The antisymmetric part B_1^- of B_1 is a biderivation, i.e. a derivation with respect to both arguments. Thus there exists a bivector $\alpha \in \Gamma(M, \bigwedge^2 TM)$ such that*

$$B_1^-(f,g) = \langle \alpha, \mathrm{d}f \wedge \mathrm{d}g \rangle,$$

for any $f, g \in A$, where $\langle\,,\rangle$ is the natural coupling between differential forms and multivectors.

This statement is a corollary of the more general Corollary 3.20 on p. 60, but we provide a direct proof below.

Proof. It suffices to show that B_1^- is a derivation in its first slot. If it is then it is also a derivation in its second slot, and any antisymmetric $\mathbb{R}$-bilinear map which is a biderivation is of the form described above. Thus, let $f, g, h \in A$. As B_1 satisfies equation (3.3), so does its antisymmetric part B_1^-. Now, using (3.3) twice (with the substitutions $f \to h$, $h \to g$ and $g \to f$ in the second line), we have

$$\begin{aligned}B_1^-(fg,h) &= fB_1^-(g,h) + B_1^-(f,gh) - B_1^-(f,g)h, \\ -B_1^-(h,fg) &= hB_1^-(f,g) - B_1^-(hf,g) - B_1^-(h,f)g.\end{aligned}$$

Then

$$\begin{aligned}2B_1^-(fg,h) &= B_1^-(fg,h) - B_1^-(h,fg) \\ &= fB_1^-(g,h) + B_1^-(f,gh) - B_1^-(hf,g) - B_1^-(h,f)g.\end{aligned}$$

But, once again using (3.3), the inner two terms can be rewritten as follows:

$$B_1^-(f,hg) - B_1^-(fh,g) = -fB_1^-(h,g) + B_1^-(f,h)g,$$

so that

$$\begin{aligned} 2B_1^-(fg,h) &= fB_1^-(g,h) - fB_1^-(h,g) + B_1^-(f,h)g - B_1^-(h,f)g \\ &= 2(fB_1^-(g,h) + gB_1^-(f,h)). \end{aligned}$$

□

Let us denote for any $f,g \in A$

$$\{f,g\}_\star := \frac{f\star g - g\star f}{\hbar} \bmod \hbar = 2B_1^-(f,g).$$

Proposition 3.7. *The bracket $\{\,,\}_\star$ satisfies the Jacobi identity,*

$$\{f,\{g,h\}_\star\}_\star + \{g,\{h,f\}_\star\}_\star + \{h,\{f,g\}_\star\}_\star = 0.$$

Proof. By a direct verification:

$$\begin{aligned} &\{f,\{g,h\}_\star\}_\star + \{g,\{h,f\}_\star\}_\star + \{h,\{f,g\}_\star\}_\star \\ &= \frac{1}{\hbar^2}\Big(f\star(g\star h) - f\star(h\star g) - (g\star h)\star f + (h\star g)\star f \\ &\qquad g\star(h\star f) - g\star(f\star h) - (h\star f)\star g + (f\star h)\star g \\ &\qquad h\star(f\star g) - h\star(g\star f) - (f\star g)\star h + (g\star f)\star h\Big) \bmod \hbar \\ &= 0. \end{aligned}$$

□

The previous two propositions show that the bracket $\{\,,\}_\star$ induced by B_1^- is in fact a Poisson bracket on M, for any star product $\star$ on M. Then it may happen that if M is a Poisson manifold with Poisson bracket $\{\,,\}$, and $\star$ is a star product on M, that the Poisson bracket of M coincides with the one induced by $\star$; that is, $\{f,g\} = \{f,g\}_\star$ for any $f,g \in A$. When this is the case, we say that the star product $\star$ is a *deformation* of the Poisson bracket $\{\,,\}$. A natural question is then whether every Poisson bracket can be deformed to a star product in this sense; Kontsevich showed in [Kon03] that this is indeed the case.

Example 3.8. (See, e.g., [Kon03] or [CI], from which the calculation below comes.) Let $M = \mathbb{R}^n$ with the Poisson bivector $\alpha = \alpha^{ij}\partial_i \wedge \partial_j$, where the coefficients α^{ij} are constant and such that $\alpha^{ij} = -\alpha^{ji}$. The bracket associated to this bivector then automatically satisfies the Jacobi identity so that it is a Poisson bracket. There is a well-known star

product, called the *Moyal product*, associated to this Poisson structure:

$$\begin{aligned}(f\star g)(x) &= \left(fg+\hbar\alpha^{ij}\partial_i(f)\partial_j(g)+\frac{\hbar^2}{2}\alpha^{ij}\alpha^{kl}\partial_i\partial_k(f)\partial_j\partial_l(g)+\cdots\right)(x)\\ &= \exp\left(\hbar\alpha^{ij}\frac{\partial}{\partial x^i}\frac{\partial}{\partial y^j}\right)f(x)g(y)\Big|_{x=y}.\end{aligned} \tag{3.4}$$

It is clear that the antisymmetric part of the first-order term B_1 of this star product equals the bracket given by $\alpha = \alpha^{ij}\partial_i\wedge\partial_j$. Moreover, this star product is associative. Indeed, observe first that

$$\frac{\partial}{\partial x}(f(x,y)|_{x=y}) = \left(\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)f(x,y)\right)\Big|_{x=y}.$$

Now

$$\begin{aligned}((f\star g)\star h)(x) &= \left[\exp\left(\hbar\alpha^{ij}\frac{\partial}{\partial x^i}\frac{\partial}{\partial z^j}\right)(f\star g)(x)h(z)\right]\Big|_{x=z}\\ &= \left[\exp\left(\hbar\alpha^{ij}\frac{\partial}{\partial x^i}\frac{\partial}{\partial z^j}\right)\left(\exp\left(\hbar\alpha^{kl}\frac{\partial}{\partial x^k}\frac{\partial}{\partial y^k}\right)f(x)g(y)\right)\Big|_{x=y}h(z)\right]\Big|_{x=z}\\ &= \left[\exp\left(\hbar\alpha^{ij}\left(\frac{\partial}{\partial x^i}+\frac{\partial}{\partial y^i}\right)\frac{\partial}{\partial z^j}\right)\exp\left(\hbar\alpha^{kl}\frac{\partial}{\partial x^k}\frac{\partial}{\partial y^k}\right)f(x)g(y)h(z)\right]\Big|_{x=y=z}\\ &= \left[\exp\left(\hbar\left(\alpha^{ij}\frac{\partial}{\partial x^i}\frac{\partial}{\partial z^j}+\alpha^{kl}\frac{\partial}{\partial y^k}\frac{\partial}{\partial z^l}+\alpha^{mn}\frac{\partial}{\partial x^m}\frac{\partial}{\partial y^n}\right)\right)f(x)g(y)h(z)\right]\Big|_{x=y=z}\\ &= \left[\exp\left(\hbar\alpha^{ij}\frac{\partial}{\partial x^i}\left(\frac{\partial}{\partial y^j}+\frac{\partial}{\partial z^j}\right)\right)\exp\left(\hbar\alpha^{kl}\frac{\partial}{\partial y^k}\frac{\partial}{\partial z^l}\right)f(x)g(y)h(z)\right]\Big|_{x=y=z}\\ &= (f\star(g\star h))(x).\end{aligned}$$

For later reference, when $M=\mathbb{R}^2$ and the Poisson bivector with respect to the coordinates x^1 and x^2 is written in the form $\alpha = \partial_1\wedge\partial_2 - \partial_2\wedge\partial_1$, then we can use the binomial theorem on (3.4) to obtain

$$\begin{aligned}(f\star g)(x) &= \exp\hbar\left(\frac{\partial}{\partial x^1}\frac{\partial}{\partial y^2}-\frac{\partial}{\partial x^2}\frac{\partial}{\partial y^1}\right)f(x)f(y)\Big|_{x=y}\\ &= \sum_{n=0}^{\infty}\frac{\hbar^n}{n!}\sum_{i=0}^{n}(-)^i\frac{\partial^{n-i}}{(\partial x^1)^{n-i}}\frac{\partial^i}{(\partial x^2)^i}f(x)\frac{\partial^i}{(\partial x^1)^i}\frac{\partial^{n-i}}{(\partial x^2)^{n-i}}g(x).\end{aligned} \tag{3.5}$$

3.3 The Kontsevich star product

In this section we give a brief description of the Kontsevich star product. We first introduce a set G_n of oriented labeled graphs.

Definition 3.9. For an integer $n \geq 0$ we say that a graph Γ belongs to G_n if

- Γ is oriented, labeled, and has no multiple edges and no loops;
- Γ has $n+2$ vertices called $\{1, \ldots, n\} \cup \{L, R\}$, and $2n$ edges;
- From each vertex k start two edges, that we denote by i_k and j_k respectively. These vertices are ordered as (i_k, j_k).

The set G_n is finite; it contains $(n(n+1))^n$ elements for $n \geq 1$ and one element for $n = 0$. It follows immediately from the definition that all edges start at the vertices $\{1, \ldots, n\}$ and none of them start at L or R.

Next, we associate to each graph $\Gamma \in G_n$ a bidifferential operator

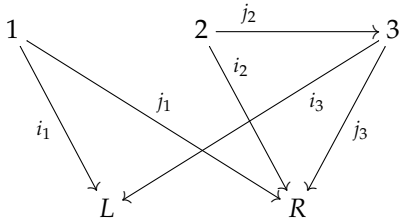
$$B_{\Gamma,\alpha} \colon A \times A \to A, \quad A = C^\infty(V), \quad V \text{ is a coordinate chart in } \mathbb{R}^n,$$

that depends on the bivector α (which need not necessarily be Poisson for this procedure), as follows:

- At the vertex k which has outgoing arrows i_k, j_k, put the coefficient $\alpha^{i_k j_k}$;
- Put a derivative $\partial_{i_k} = \partial/\partial x^{i_k}$ at each edge i_k, and ∂_{j_k} at each edge j_k;
- Place a function $f \in A$ at L and $g \in A$ at R;
- Multiply all elements described above in the order given by the labeling of the graph.

Take for example the following graph:

$\Gamma \; =$



This graph corresponds to the differential operator

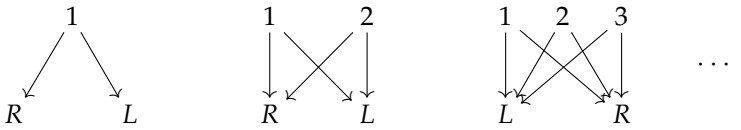
$$B_{\Gamma,\alpha} = \alpha^{i_1 j_i} \alpha^{i_2 j_2} \partial_{j_2}(\alpha^{i_3 j_3}) \partial_{i_1} \partial_{i_3}(f) \partial_{j_1} \partial_{i_2} \partial_{j_3}(g).$$

Kontsevich then essentially proved in [Kon03] that there exist weights $w_\Gamma \in \mathbb{R}$ for each graph $\Gamma \in G_n$ that do not depend on the bivector α, such that the formula

$$f \star g := \sum_{n=0}^{\infty} \hbar^n \sum_{\Gamma \in G_n} w_\Gamma B_{\Gamma,\alpha}(f, g) \tag{3.6}$$

defines a star product in the sense of Definition 3.4. There is an explicit formula for the weights w_Γ included in [Kon03], but for brevity we do not include it here.

Example 3.10. Returning to the Moyal product on $M = \mathbb{R}^n$ (c.f. Example 3.8), the only nonzero operators $B_{\Gamma,\alpha}$ that survive in the sum (3.6) are the ones corresponding to the graphs (suppressing the labels i_k, j_k of the edges for clarity)



(3.7)

That is, all arrows land on L and R (the placeholders of the functions f and g), and none of them land on the vertices $\{1, \dots, n\}$, as a consequence of the fact that the coefficients α^{ij} of the Poisson bivector α are constant. Then one can show that the weights w_Γ of the surviving graphs (3.7) are such that the Kontsevich star product (3.6) reduces precisely to the Moyal star product (3.4).

3.4 Gauge transformations

The algebra $A[\![\hbar]\!]$ is an $\mathbb{R}[\![\hbar]\!]$-module, and as such, we may consider its group of automorphisms; that is, invertible maps $D\colon A[\![\hbar]\!] \to A[\![\hbar]\!]$ which are linear over $\mathbb{R}[\![\hbar]\!]$. For $f \in A$, such an automorphism D acting on f is necessarily of the form

$$D(f) = D_0(f) + \sum_{n>0} \hbar^n D_n(f)$$

for certain $\mathbb{R}[\![\hbar]\!]$-linear operators D_n. How D acts on arbitrary elements is again given by $\mathbb{R}[\![\hbar]\!]$-linearity:

$$D\left(\sum_{n\geq 0} \hbar^n f_n\right) = \sum_{n\geq 0} \hbar^n D(f_n) = \sum_{n\geq 0, m\geq 0} \hbar^{n+m} D_m(f_n).$$

Definition 3.11. Let $D = \sum_{n\geq 0} \hbar^n D_n\colon A[\![\hbar]\!] \to A[\![\hbar]\!]$ be an automorphism of $\mathbb{R}[\![\hbar]\!]$-modules. If D_1 equals the identity while D_n for $n > 0$ is a differential operator of bounded order, then we say that D is a *gauge transformation*.

The set of gauge transformations has a natural action on star products. If D is a gauge transformation, this action on a star product $\star$ is defined as follows:[3]

$$f \star' g := D^{-1}(D(f) \star D(g)).$$

[3]This also explains why the zeroth-order term of D must be the identity: if it were something else, say D_0, then we would have $f \star' g = D_0^{-1}(D_0(f)D_0(g)) + O(\hbar)$. However, by the definition of star products (see Definition 3.4 on p. 48), the zeroth order term of $f \star' g$ must be the pointwise product of f and g, so that this formula would imply that $fg = D_0^{-1}(D_0(f)D_0(g))$. This can impossibly hold if D_0 is a differential operator different from the identity.

In other words, $\star'$ is the unique star product that makes D into an algebra homomorphism (here D^{-1} must be understood as the inverse in the sense of formal power series; note that D^{-1} exists since D is assumed to be an automorphism). Whenever two star products are related to each other via such an automorphism we say that they are *gauge equivalent*, or just equivalent for short.[4]

Since D^{-1} is also an $\mathbb{R}[\![\hbar]\!]$-linear automorphism, it too can be expanded as a series in $\hbar$:

$$D^{-1} = 1 + \sum_{n>0} \widetilde{D}_n$$

for certain differential operators $\widetilde{D}_n$ (note that these are *not* such that $\widetilde{D}_n = D_n^{-1}$). Necessarily, these operators $\widetilde{D}_n$ are completely determined by the demand that $D \circ D^{-1} = D^{-1} \circ D = 1$.

Examining the equation $f = D^{-1}(D(f))$ up to first order, we can explicitly determine $\widetilde{D}_1$:

$$\begin{aligned} f &= D^{-1}(f + \hbar D_1(f) + O(\hbar^2)) \\ &= D^{-1}(f) + D^{-1}(\hbar D_1(f)) + O(\hbar^2) \\ &= f + \hbar \widetilde{D}_1(f) + \hbar D_1(f) + O(\hbar^2) \end{aligned}$$

so that we find $\widetilde{D}_1(f) = -D_1(f)$. Using this, let us now calculate how a gauge transformation D acts on the first order term of star products: suppose that D maps $\star$ to $\star'$. Then

$$\begin{aligned} f \star' g &= D^{-1}(Df \star Dg) \\ &= (1 - \hbar D_1 + O(\hbar^2)) \left((f + \hbar D_1 f + O(\hbar^2)) \star (g + \hbar D_1 g + O(\hbar^2)) \right) \\ &= (1 - \hbar D_1 + O(\hbar^2)) \left(fg + \hbar (f D_1 g + g D_1 f + B_1(f,g)) + O(\hbar^2)) \right) \\ &= fg + \hbar (B_1(f,g) + f D_1 g + g D_1 f - D_1(fg)) + O(\hbar^2) \end{aligned}$$

so that the first order term B_1' of $\star'$ equals

$$B_1'(f,g) = B_1(f,g) + f D_1 g + g D_1 f - D_1(fg). \tag{3.8}$$

Since the latter three terms are symmetric under an exchange of f and g, it follows immediately that the antisymmetric part does not change, i.e., $(B_1')^-(f,g) = B_1^-(f,g)$. This has the following immediate consequence.

Proposition 3.12. *Let M be a Poisson manifold with Poisson bracket $\{\,,\}$. If $\star$ is a deformation of the Poisson bracket $\{\,,\}$, then all star products that are equivalent to $\star$ are also deformations of $\{\,,\}$.*

[4] When $\star$ and $\star'$ are such that the corresponding bilinear operators B_n and B_n' are differential operators, and D maps $\star$ to $\star'$, then one can prove that it follows from this that each D_n is a linear differential operator of bounded order; see for example [GR99] for a proof of this.

As to the positive part of B_1, we have the following (which we will prove later).

Proposition 3.13. *For any star product with first order term B_1 there is a gauge transformation D such that B_1' is skew-symmetric, i.e. $(B_1')^+ = 0$.*

This allows us to restrict the search for deformations of a Poisson structure to those that have no symmetric first-order part. Equation (3.8) and the observation directly below it show that for this to hold, $D = 1 + \sum_n \hbar^n D_n$ must be such that

$$B_1^+(f,g) = D_1(fg) - fD_1g - gD_1f. \tag{3.9}$$

Thus, Proposition 3.13 states that there always exists a differential operator D_1 such that this holds. We remark that this equation precisely expresses that B_1^+ is a coboundary with respect to the Hochschild differential d^H; in the next section we will see that this is true (see Corollary 3.18 on p. 60), thus proving Proposition 3.13.

For later reference, let us study in more detail what a gauge transformed star product looks like. We take a star product $\star = \hbar^n B_n$; here it will be convenient to include the zeroth term $B_0 = \cdot$ in the sum as above. (We will make heavy use of the Einstein summation convention here; also, all sums start at 0 unless explicitly written otherwise.) Similarly, we write our gauge transformation as $D = \hbar^n D_n$. We first examine the inverse $D^{-1} = \hbar^i \widetilde{D}_i$. Taking $f \in A[\![\hbar]\!]$, we must have

$$f = D^{-1}D(f) = \hbar^i \widetilde{D}_i \left(\hbar^j D_j(f)\right) = \sum_{n=i+j} \hbar^n \widetilde{D}_i D_j(f).$$

For this to hold all terms of order $n > 0$ must be zero. Recalling that $D_0 = \mathrm{Id}$, we can thus express $\widetilde{D}_n$ inductively as

$$\widetilde{D}_n = -D_n - \sum_{i=1}^{n-1} \widetilde{D}_i D_{n-i}. \tag{3.10}$$

Next, taking $f, g \in A[\![\hbar]\!]$, we examine the following expression:

$$\begin{aligned} Df \star Dg &= \left(\hbar^i D_i(f)\right) \star \left(\hbar^j D_j(g)\right) = \hbar^k B_k \left(\hbar^i D_i(f), \hbar^j D_j(g)\right) \\ &= \sum_{n=i+j+k} \hbar^n B_k \left(D_i(f), D_j(g)\right). \end{aligned}$$

Using now that $D_0 = \mathrm{Id}$ and $B_0 = \cdot$, we write this schematically as

$$Df \star Dg = \sum_n \hbar^n \Big(B_n(f,g) + gD_nf + fD_ng + \text{terms involving } B_{<n} \text{ and } D_{<n}\Big).$$

Finally, using (3.10) on the above we conclude for the gauge transformed star product $\star'$

that

$$f \star' g = D^{-1}(Df \star Dg) \tag{3.11}$$
$$= \sum_n \hbar^n \Big(B_n(f,g) + gD_nf + fD_ng - D_n(fg) + \text{terms involving } B_{<n} \text{ and } D_{<n} \Big).$$

3.5 Hochschild cohomology

In the previous sections we have seen two hints of the Hochschild cohomology, for a good reason: the subject of star products and deformation quantization is intimately connected to the Hochschild cohomology of the algebra A. In this section we explore (part of) this connection.

Definition 3.14. A *differential graded Lie algebra* $\mathfrak{g}$ over a field k of characteristic zero, or DGLA for short, is a graded vector space $\mathfrak{g} = \oplus_i \mathfrak{g}_i$ over k, together with a bilinear map $[\,,\,]\colon \mathfrak{g}_i \times \mathfrak{g}_j \to \mathfrak{g}_{i+j}$ and a differential $\mathrm{d}\colon \mathfrak{g}_i \to \mathfrak{g}_{i+1}$ satisfying the following for any three homogeneous elements $x, y, z \in L$:

$$[x,y] = -(-)^{|x||y|}[y,x], \qquad \text{(skew-symmetry)}$$
$$[x,[y,z]] = [[x,y],z] + (-)^{|x||y|}[y,[x,z]], \qquad \text{(graded Jacobi identity)}$$
$$\mathrm{d}([x,y]) = [\mathrm{d}x, y] + (-1)^{|x|}[x, \mathrm{d}y]. \qquad \text{(graded Leibniz rule)}$$

We denote the shifted space of multivector fields with $T^k_{\text{multi}}(M) = \Gamma(\bigwedge^{k+1} TM)$, and write

$$T_{\text{multi}}(M) = \bigoplus_{k=-1}^{\infty} T^k_{\text{multi}}(M)$$

for the sum; equipping $T_{\text{multi}}(M)$ with the zero differential $\mathrm{d}^{\wedge TM} = 0$ and the Schouten bracket $[\![\,,\,]\!]$ makes it into a DGLA.

The other relevant DGLA is called the *Hochschild DGLA* of $A = C^\infty(M)$:

$$C(A) = \bigoplus_{k=-1}^{\infty} C^k(A), \qquad C^k(A) = \mathrm{Hom}_{\mathbb{R}}(A^{\otimes(k+1)}, A),$$

endowed with the Hochschild differential $\mathrm{d}^H\colon C^k(A) \to C^{k+1}(A)$ given by, for $\Phi \in C^k(A)$,

$$\begin{aligned} \mathrm{d}^H\Phi(f_0 \otimes \cdots \otimes f_{k+1}) = {} & f_0\Phi(f_1 \otimes \cdots \otimes f_{k+1}) \\ & - \sum_{i=0}^{k} (-)^i \Phi(f_0 \otimes \cdots \otimes f_i f_{i+1} \otimes \cdots \otimes f_{k+1}) \\ & + (-)^k \Phi(f_0 \otimes \cdots \otimes f_k) f_{k+1}. \end{aligned} \tag{3.12}$$

Although it will not be important for our purposes, for completeness we also include the definition of the bracket of this DGLA. For $\Phi_i \in C^{k_i}(A)$ it is given by

$$[\Phi_1, \Phi_2] = \Phi_1 \circ \Phi_2 - (-)^{k_1 k_2} \Phi_2 \circ \Phi_1,$$

where the (non-associative) product $\circ$ is defined by

$$\begin{aligned}&(\Phi_1 \circ \Phi_2)(f_0 \otimes \cdots \otimes f_{k_1+k_2}) \\ &= \sum_{i=0}^{k_1} (-1)^{ik_2} \Phi_1(f_0 \otimes \cdots \otimes f_{i-1} \otimes (\Phi_2(f_i \otimes \cdots \otimes f_{i+k_2})) \otimes f_{i+k_2+1} \otimes \cdots \otimes f_{k_1+k_2}).\end{aligned}$$

Writing $m_A \colon A \times A \to A$ for the pointwise product $m_A(f,g) = fg$ on A, note that with respect to this bracket, the Hochschild differential d^H can be expressed concisely as

$$\mathrm{d}^H = \operatorname{ad} m_A = [m_A, \cdot].$$

Using this expression and the graded Jacobi for $[\,,\,]$ it is easy to prove that d^H is indeed a differential:

$$\mathrm{d}^H(\mathrm{d}^H\Phi) = [m_A, [m_A, \Phi]] = [[m_A, m_A], \Phi] - [m_A, [m_A, \Phi]] = -[m_A, [m_A, \Phi]]$$

from which it follows that $(\mathrm{d}^H)^2 = 0$. The graded Leibniz rule is similarly easy:

$$\begin{aligned}\mathrm{d}^H[\Phi_1, \Phi_2] &= [m_A, [\Phi_1, \Phi_2]] = [[m_A, \Phi_1], \Phi_2] + (-)^{|m_A||\Phi_1|}[\Phi_1, [m_A, \Phi_2]] \\ &= [\mathrm{d}^H\Phi_1, \Phi_2] + (-)^{|\Phi_1|}[\Phi_1, \mathrm{d}^H\Phi_2].\end{aligned}$$

That the bracket $[\,,\,]$ is skew-symmetric follows immediately from its definition. The only thing that remains in order to show that $(C(A), \mathrm{d}^H, [\,,\,])$ is indeed a DGLA is the graded Jacobi identity for the bracket $[\,,\,]$; for this we refer to, for example, [CI].

There is an important map $\mathcal{U} \colon T^k_{\text{multi}}(M) \to C^k(A)$, which is an extension of the usual identification between vector fields and first order differential operators. This map is defined as follows: if $\chi = X_0 \wedge \cdots \wedge X_k$ is a homogeneous multivector field then

$$\mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_k) = \frac{1}{(k+1)!} \sum_{\sigma \in S_{k+1}} (-)^\sigma X_{\sigma(0)}(f_0) \cdots X_{\sigma(k)}(f_k), \tag{3.13}$$

and it is of course linearly extended to act on all of $T^k_{\text{multi}}(M)$. The definition is extended to T^0_{multi}, i.e. vector fields, as the identity map.

Proposition 3.15. *The map $\mathcal{U}$ is a chain map:*

$$\mathrm{d}^H\mathcal{U}(\chi) = 0 = \mathcal{U}(\mathrm{d}^{\wedge TM}\chi).$$

Proof. Applying equation (3.12) to $\mathrm{d}^H\mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_{k+1})$, consider the i-th term in

the sum:

$$\mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_i f_{i+1} \otimes \cdots \otimes f_k) = f_i \mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_{i+1} \otimes \cdots \otimes f_k) + \mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_i \otimes \cdots \otimes f_k) f_{i+1},$$

which holds due to the Leibniz rule that $X_{\sigma(i)}$ satisfies (it being a vector field). But now consider the $(i+1)$-th term:

$$\mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_{i+1} f_{i+2} \otimes \cdots \otimes f_k) = f_{i+1} \mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_{i+2} \otimes \cdots \otimes f_k) + \mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_{i+1} \otimes \cdots \otimes f_k) f_{i+2}.$$

The first term on the right hand side in fact equals the second term of the right hand side of the equation above it, and since the signs of the corresponding terms in equation (3.13) are always opposite these two terms cancel against each other. Thus, each term of the sum in (3.13) splits into two terms, where the right term always cancels against the left term of the next term of the sum. The only terms that are left after this are

$$-f_0 \mathcal{U}(\chi)(f_1 \otimes \cdots \otimes f_k) - (-)^k \mathcal{U}(\chi)(f_0 \otimes \cdots \otimes f_{k-1}) f_k,$$

but these cancel against the first and last terms of the right hand side of the defining equation of the Hochschild differential (3.12). □

Defining the *Hochschild cohomology*,

$$HH^k(A) = \frac{\ker(\mathrm{d}^H \colon C^k(A) \to C^{k+1}(A))}{\mathrm{im}(\mathrm{d}^H \colon C^{k-1}(A) \to C^k(A))},$$

the proposition above implies that the map $\mathcal{U}$ descends to the Hochschild cohomology:

$$\widetilde{\mathcal{U}} \colon T^k_{\text{multi}}(M) \to HH^k(A), \quad \widetilde{\mathcal{U}}(\chi) := [\mathcal{U}(\chi)].$$

(Note that T^k_{multi} coincides with its own cohomology because its differential is 0).

Theorem 3.16 (Hochschild-Kostant-Rosenberg). *$\widetilde{\mathcal{U}}$ is an isomorphism of vector spaces,*

$$T^k_{\text{multi}}(M) \cong HH^k(A).$$

This version of the HKR-theorem is proved explicitly in [Kon03, section 4.6].

For any $\lambda \in C^k(A)$ let λ^Σ denote its antisymmetrization, i.e.

$$\lambda^\Sigma(f_0, \ldots, f_k) = \frac{1}{(k+1)!} \sum_{\sigma \in S_{k+1}} (-)^\sigma \lambda(f_{\sigma(0)}, \ldots, f_{\sigma(k)}). \tag{3.14}$$

In the next lemma we show that the image of the Hochschild differential d^H has no antisymmetrization. Combined with the HKR theorem, this observation will allow us to gain a thorough understanding of the relationships between $C^k(A)$, T^k_{multi} and d^H; in particular, it will allow us to prove Proposition 3.13 on p. 55.

Lemma 3.17. *For all $\lambda \in C^k(A)$ we have*

$$(\mathrm{d}^H\lambda)^\Sigma = 0.$$

Proof. For any $O \in C^k(A)$, let us define a shorthand notation: if $\sigma \in S_{k+1}$ then

$$\sigma\,[O(f_0,\dots,f_k)] = O(f_{\sigma(0)},\dots,f_{\sigma(k)}).$$

Expanding $(\mathrm{d}^H\lambda)^\Sigma$, we obtain

$$\begin{aligned}(\mathrm{d}^H\lambda)^\Sigma(f_0\otimes\cdots\otimes f_{k+1}) = \sum_{\sigma\in S_{k+1}} (-)^\sigma\Big(&\sigma\Big[f_0\lambda(f_1\otimes\cdots\otimes f_{k+1})\Big]\\ &-\sum_{i=0}^{k}(-)^i\sigma\Big[\lambda(f_0\otimes\cdots\otimes f_if_{i+1}\otimes\cdots\otimes f_{k+1})\Big]\\ &+(-)^k\sigma\Big[\lambda(f_0\otimes\cdots\otimes f_k)f_{k+1}\Big]\Big).\end{aligned} \tag{3.15}$$

Take a $\sigma \in S_{k+1}$ and consider the first term:

$$(-)^\sigma\sigma\Big[f_0\lambda(f_1\otimes\cdots\otimes f_{k+1})\Big]. \tag{3.16}$$

Now we take $\tau = \begin{pmatrix}0 & 1 & \cdots & k & k+1\end{pmatrix} \in S_{k+1}$, whose sign is $-(-)^k$, and we consider the last term of the right hand side of (3.15) with respect to the element $\sigma\tau \in S_{k+1}$:

$$(-)^{\sigma\tau}(-)^k\sigma\tau\Big[\lambda(f_0\otimes\cdots\otimes f_k)f_{k+1}\Big] = -(-)^k(-)^\sigma(-)^k\sigma\Big[\lambda(f_1\otimes\cdots\otimes f_{k+1})f_0\Big]$$

which is minus the expression in (3.16). As σ and $\sigma\tau$ both occur in the sum $\sum_{\sigma\in S_{k+1}}$ over S_{k+1}, then, these two terms cancel against each other. In this way, for each first term of the right hand side of (3.15), there is a last term in the right hand side in (3.15) that cancels it. Therefore, none of the first and last terms survive.

The middle terms of (3.15) vanish for a similar reason: the term

$$(-)^\sigma\sigma\Big[\lambda(f_0\otimes\cdots\otimes f_if_{i+1}\otimes\cdots\otimes f_{k+1})\Big]$$

cancels against the term (setting $\rho = \begin{pmatrix}i & i+1\end{pmatrix}$)

$$\begin{aligned}(-)^{\sigma\rho}\sigma\rho\Big[\lambda(f_0\otimes\cdots\otimes f_if_{i+1}\otimes\cdots\otimes f_{k+1})\Big]\\ = -(-)^\sigma\sigma\Big[\lambda(f_0\otimes\cdots\otimes f_{i+1}f_i\otimes\cdots\otimes f_{k+1})\Big]\end{aligned}$$

which also appears in the sum over S_{k+1}. □

Note that for $k = 0$, i.e., for $\lambda\colon A \to A$, the statement that $(\mathrm{d}^H\lambda)^\Sigma = 0$ is the same as

saying that $\mathrm{d}^H\lambda\colon A\otimes A\to A$ is symmetric in its two arguments. This case is particularly easy to see when writing it out:

$$\mathrm{d}^H\lambda(f,g)=f\lambda(g)-\lambda(fg)+\lambda(f)g=\mathrm{d}^H\lambda(g,f).$$

Corollary 3.18. *A cocycle $B\in C^k(A)$ is exact if and only if $B^\Sigma=0$.*

Proof. The Lemma above already proved one direction; therefore we only need to prove that $B^\Sigma=0$ implies exactness. Suppose therefore that $B^\Sigma=0$. Since $\widetilde{\mathcal{U}}$ is an isomorphism by the HRK Theorem 3.16, there exists a $\chi\in T^k_{\text{multi}}(M)$ such that $[B]=\widetilde{\mathcal{U}}(\chi)=[\mathcal{U}(\chi)]$. This is just another way of saying

$$B=\mathcal{U}(\chi)+\mathrm{d}^H\lambda\qquad\text{for some }\lambda\in C^{k-1}(A).$$

Moving $\mathrm{d}^H\lambda$ to the other side and taking the antisymmetrization of both sides, we obtain

$$(B-\mathrm{d}^H\lambda)^\Sigma=0+0=\mathcal{U}(\chi)^\Sigma=\mathcal{U}(\chi);$$

the last equality holds as a direct consequence of the definition (3.13) of $\mathcal{U}$. Thus $B=\mathrm{d}^H\lambda$. □

This finally shows that equation (3.9) on p. 55 holds and with that Proposition 3.13 is proven.

Corollary 3.19. *Each equivalence class $[B]$ of a cocycle $B\in C^k(A)$ has a unique totally antisymmetric representative that is of the form $U(\chi)$ for a certain $\chi\in T^k_{\text{multi}}(M)$. That is, any cocycle B can be written as*

$$B=U(\chi)+\mathrm{d}^H\lambda$$

for a certain $\lambda\in C^{k-1}(A)$.

Proof. Again by the HKR theorem, there must exist a $\chi\in T^k_{\text{multi}}(M)$ such that $[B]=[\mathcal{U}(\chi)]$; as $\mathcal{U}(\chi)$ is completely antisymmetric by definition, this proves existence. As to uniqueness, let $B,B'\in[B]$, so that $B'=B+\mathrm{d}^H\lambda$ for some $\lambda\in C^{k-1}(A)$. Suppose that they are both completely antisymmetric. Then

$$B'=(B')^\Sigma=(B+\mathrm{d}^H\lambda)^\Sigma=B^\Sigma=B.$$ □

Corollary 3.20. *Any totally antisymmetric cocycle $B\in C^k(A)$ satisfies $B=\mathcal{U}(\chi)$ for a certain $\chi\in T^k_{\text{multi}}(M)$.*

Corollary 3.21. *If $B\in C^k(A)$ is d^H-closed and completely antisymmetric then it is a derivation with respect to all of its arguments.*

Corollary 3.22. *Let $B \in C^k(A)$ be d^H-closed. Let χ be such that $B^\Sigma = \mathcal{U}(\chi)$. Then $\widetilde{\mathcal{U}}^{-1}([B]) = \chi$.*

Proof. Write $B = B^\Sigma + (B - B^\Sigma) =: B^\Sigma - \widetilde{B}$, then clearly $\widetilde{B}^\Sigma = 0$. Thus $\widetilde{B} = \mathrm{d}^H\lambda$ for a certain λ and so

$$[B] = [\mathcal{U}(\chi)] + [\mathrm{d}^H\lambda] = [\mathcal{U}(\chi)] = \widetilde{\mathcal{U}}(\chi),$$

from which the result follows. □

3.5.1 The star product up to order 2

In section 1.4.2 in [Kon03], Kontsevich writes out explicitly a formula for a star product up to and including order 2:

$$\begin{aligned} f \star g = fg + \hbar\alpha^{ij}\partial_i(f)\partial_j(g) + \frac{\hbar^2}{2}\alpha^{ij}\alpha^{kl}\partial_i\partial_k(f)\partial_j\partial_l(g) \\ + \frac{\hbar^2}{3}\left(\alpha^{ij}\partial_j(\alpha^{kl})(\partial_i\partial_k(f)\partial_l(g) - \partial_k(f)\partial_i\partial_l(g))\right) + O(\hbar^3). \end{aligned} \tag{3.17}$$

However, this does not seem to agree with his general formula (3.6), because terms of the following form do not occur in it:

$$\Gamma \;=\; \begin{array}{ccc} 1 & \rightleftarrows & 2 \\ \downarrow & & \downarrow \\ L & & R \end{array} \qquad \text{with corresponding operator} \quad B_{\Gamma,\alpha} = \partial_k\alpha^{ij}\partial_i\alpha^{kl}\partial_j \otimes \partial_l. \tag{3.18}$$

In this section we explain why these terms are absent from Kontsevich's formula: they will be gauge transformed away by a suitable gauge transformation D.

Proposition 3.23. *Any d^H-exact term can be gauge-transformed away from any star product $\star = 1 + \sum_{n>0}\hbar^n B_n$. That is, if B_n contains a d^H-exact term $\mathrm{d}^H\lambda$ for some n, then $\star$ is gauge-equivalent to a star product $\star' = 1 + \sum_{n>0}\hbar^n B'_n$ such that*

- *$B'_m = B_m$ for $m < n$,*
- *$B'_n = B_n - \mathrm{d}^H\lambda$.*

Proof. Suppose that $B_n(f,g) = P(f,g) + \mathrm{d}^H\lambda(f,g)$ for certain $P \in C^1(A)$ and $\lambda \in C^0(A)$. Note that we can rewrite equation (3.11) as

$$B'_n(f,g) = B_n(f,g) + \mathrm{d}^H D_n(f,g) + \text{terms involving } B_{<n} \text{ and } D_{<n}.$$

Now we set $D_m = 0$ for $m \neq n$ and $D_n = -\lambda$. Then the terms involving $B_{<n}$ and $D_{<n}$ vanish, and we find

$$B'_n(f,g) = B_n(f,g) - \mathrm{d}^H\lambda(f,g) = P(f,g) + \mathrm{d}^H\lambda(f,g) - \mathrm{d}^H\lambda(f,g) = P(f,g),$$
$$B'_m(f,g) = B_m(f,g) \qquad \text{when } m < n.$$

Note that $D = 1 - \hbar^n\lambda$ will occur in B'_m when $m > n$; therefore, only the terms of $\star$ of order lower than n remain unmodified. If it happens that B'_m for $m > n$ contains d^H-exact terms, hoewever, then these can be removed by this very procedure. □

Now the term in question $B_{\Gamma,\alpha}$ from (3.18) is d^H-exact. Indeed,

$$B_{\Gamma,\alpha} = \mathrm{d}^H\left(-\frac{1}{2}\partial_k(\alpha^{ij})\partial_i(\alpha^{kl})\partial_j\partial_l\right).$$

Therefore, we can remove it by using the proposition above. Thus, the formula (3.17) does *not* equal the general formula (3.6) up to order 2; instead, they are gauge equivalent.

3.6 The Kontsevich star product on the dual of Lie algebras

Consider the dual $\mathfrak{g}^*$ of a finite-dimensional Lie algebra $\mathfrak{g}$. Being a vector space, this dual is also a manifold, and it inherits a natural Poisson structure from the Lie bracket $[,]$ on $\mathfrak{g}$. In this section we will work out in detail what the Kontsevich star product is in this case. First we fix some notation and review basic theory.

3.6.1 The dual as a Poisson manifold

The dual $\mathfrak{g}^*$ of $\mathfrak{g}$ is the topic of this section, so we write x^i with upper indices for its coordinates, so that its basis vectors are e_i with lower indices. The basis vectors of $\mathfrak{g}$ dual to e_i are then e^i.

Remark 3.24. The basis vectors e^i of $\mathfrak{g}$ act on $\mathfrak{g}^*$ by virtue of $(\mathfrak{g}^*)^* = \mathfrak{g}$, i.e. $e^i(x^j e_j) = x^j e_j(e^i) = x^i$. As e^i is a linear function on $\mathfrak{g}^*$ it is an element of $C^\infty(\mathfrak{g}^*)$, and the previous calculation shows that in fact it coincides with the coordinate function. Therefore, we may think of the basis vectors of $\mathfrak{g}$ not only as such, but also as the coordinate functions on $\mathfrak{g}^*$.

The structure of $\mathfrak{g}^*$ as a vector space yields an isomorphism $\mathfrak{g}^* \cong T_v\mathfrak{g}^*$ for each $v \in \mathfrak{g}^*$. This isomorphism is given by the map $v \mapsto X_v$ where X_v is the unique derivation that acts as the directional derivative in the direction of v; i.e., for $f \in C^\infty(\mathfrak{g}^*)$ we have

$$X_v(f)(a) = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} f(a + tv).$$

By the chain rule it is immediate that this isomorphism maps the basis vector $e_i \in \mathfrak{g}^*$ to $\left.\frac{\partial}{\partial x^i}\right|_v \in T_v\mathfrak{g}^*$. Now consider the *co*tangent space of $\mathfrak{g}^*$, i.e. $T_v^*\mathfrak{g}^*$. Then this isomorphism

gives

$$T_v^*\mathfrak{g}^* = (T_v\mathfrak{g}^*)^* \cong (\mathfrak{g}^*)^* \cong \mathfrak{g}.$$

If e^i is the basis of $\mathfrak{g}$ dual to the basis e_i of $\mathfrak{g}^*$, then this isomorphism maps e^i to $\mathrm{d}x^i|_v$.

Definition 3.25. The Poisson bracket on $\mathfrak{g}^*$ is the unique biderivation on $C^\infty(\mathfrak{g}^*)$ whose value on $\gamma_1, \gamma_2 \in \mathfrak{g} = \mathfrak{g}^{**} \subset C^\infty(\mathfrak{g}^*)$ is $[\gamma_1, \gamma_2] \in \mathfrak{g}^{**}$. That is, writing $\langle\,,\rangle\colon \mathfrak{g}^* \times \mathfrak{g} \to \mathbb{R}$ for the coupling between $\mathfrak{g}^*$ and $\mathfrak{g}$, we have for $x \in \mathfrak{g}^*$

$$\{\gamma_1, \gamma_2\}(x) = \langle x, [\gamma_1, \gamma_2]\rangle.$$

This determines the bracket completely, because if its value on all linear functions is known then its values on the coordinate functions is known. Indeed, taking the basis e^i of $\mathfrak{g}$ and an $x \in \mathfrak{g}^*$ we may calculate its coefficients α^{ij} of the bivector $\alpha \in \Gamma(\bigwedge^2 T\mathfrak{g}^*)$ that corresponds to this Poisson structure at $x \in \mathfrak{g}^*$ by

$$\alpha^{ij}(x) = \{e^i, e^j\}(x) = \langle x, [e^i, e^j]\rangle = \langle x^l e_l, c_k^{ij} e^k\rangle = c_k^{ij} x^k.$$

To calculate what it does on arbitrary functions $f, g \in C^\infty(\mathfrak{g}^*)$, note first that $[\,,]$ descends onto $\mathfrak{g} \wedge \mathfrak{g}$ because of its antilinearity. Writing $\mathrm{d}f|_x = \frac{\partial f}{\partial x^i}(x)e^i \in \mathfrak{g}$ and similarly for g using the isomorphism described above, we find

$$\begin{aligned}\{f,g\}(x) &= \langle \alpha, \mathrm{d}f \wedge \mathrm{d}g\rangle(x) = \langle \alpha(x), \mathrm{d}f|_x \wedge \mathrm{d}g|_x\rangle = \langle x, [\mathrm{d}f|_x, \mathrm{d}g|_x]\rangle \\ &= \left\langle x^k e_k, \left[\frac{\partial f}{\partial x^i}(x)e^i, \frac{\partial g}{\partial x^j}(x)e^j\right]\right\rangle \\ &= \left\langle x^k e_k, c_l^{ij} e^l \frac{\partial f}{\partial x^i}(x)\frac{\partial g}{\partial x^j}(x)\right\rangle = c_k^{ij} x^k \frac{\partial f}{\partial x^i}(x)\frac{\partial g}{\partial x^j}(x).\end{aligned}$$

This bivector inherits the Jacobi identity from that of $[\,,]$ on $\mathfrak{g}$, so that $(\mathfrak{g}^*, \{\,,\})$ is indeed a Poisson manifold.

3.6.2 The enveloping and symmetric algebras

Denote with $U(\mathfrak{g}) = T(\mathfrak{g})/(x \otimes y - y \otimes x - [x,y])$ the enveloping algebra of the Lie algebra $\mathfrak{g}$. If $x \in \mathfrak{g}$ denote with $\overline{x}$ the corresponding element of $U(\mathfrak{g})$. If $\{e^i\}_i$ is a basis for $\mathfrak{g}$ then we have the following well-known theorem.

Theorem 3.26 (Poincaré-Birkhoff-Witt). *The algebra $U(\mathfrak{g})$ has the following basis:*

$$\overline{e^{i_1}} \cdots \overline{e^{i_n}}, \quad i_1 \leq \cdots \leq i_n, \quad n \in \mathbb{N}_{\geq 0},$$

together with 1.

In particular, the map that sends x to $\overline{x}$ is injective so that we may drop the distinction between x and $\overline{x}$; henceforth we shall denote them both with x.

The algebra $U(\mathfrak{g})$ comes with a natural filtration: namely, $U_n(\mathfrak{g})$ is the subspace of $U(\mathfrak{g})$ generated by products of at most n elements from $\mathfrak{g}$. Therefore we may consider the

associated graded space: $\operatorname{gr}_n U(\mathfrak{g}) = U_n(\mathfrak{g})/U_{n-1}(\mathfrak{g})$ and $\operatorname{gr} U(\mathfrak{g}) = \oplus_n \operatorname{gr}_n U(\mathfrak{g})$, which has a well-defined multiplication induced by that of $U(\mathfrak{g})$. π is the natural projection map $U(\mathfrak{g}) \to \operatorname{gr} U(\mathfrak{g})$ that maps $a \in U_n(\mathfrak{g})$ to $\pi(a) = a + U_{n-1}(\mathfrak{g}) \in \operatorname{gr}_n U(\mathfrak{g})$. Since $U(\mathfrak{g})$ is as an algebra generated by the elements $x \in \mathfrak{g} \subset U(\mathfrak{g})$, it follows that $\operatorname{gr} U(\mathfrak{g})$ is generated by the elements $\pi(x)$, $x \in \mathfrak{g}$. But these elements commute:

$$\pi(x)\pi(y) - \pi(y)\pi(x) = \pi(xy - yx) = \pi([x,y]) = 0 \in \operatorname{gr}_2 U(\mathfrak{g}).$$

Thus $\operatorname{gr} U(\mathfrak{g})$ is a commutative algebra. Hence, by the universal property of the symmetric algebra $S(\mathfrak{g})$ there exists an algebra homomorphism

$$I\colon S(\mathfrak{g}) \to \operatorname{gr} U(\mathfrak{g})$$

that extends the linear map $\mathfrak{g} \to \operatorname{gr} U(\mathfrak{g})$ given by $x \mapsto \pi(x)$. Since the latter elements generate $\operatorname{gr} U(\mathfrak{g})$, this map is surjective.

Proposition 3.27. *The PBW theorem is equivalent with the following statement: the map $I\colon S(\mathfrak{g}) \to \operatorname{gr} U(\mathfrak{g})$ is an isomorphism of algebras.*

Proof. It suffices to show that I is injective. Set $x_J = x_{j_1} \cdots x_{j_n}$, where J is an increasing multi-index of length n. Since $\operatorname{gr}_n U(\mathfrak{g})$ is commutative, as a vector space it is spanned by elements of the form $\pi(x_J)$. These are the image under I of the monomial basis in $S^n(\mathfrak{g})$. Now I is injective if and only if the $\pi(x_J)$ are a basis in $\operatorname{gr}_n U(\mathfrak{g})$, i.e. there is no relation of the form

$$0 = \sum_{|J|=n} c_J \pi(x_J) = \sum_{|J|=n} c_J x_J + U_{n-1}$$

which is the same as saying that there is no relation of the form

$$0 = \sum_{|J|\le n} c_J x_J;$$

note the $\le$ sign in the sum. This is equivalent with the monomials x_J with $|J| \le n$ being linearly independent in $U_n(\mathfrak{g})$, which is the PBW theorem. That these monomials also span $U_n(\mathfrak{g})$ can be proved easily using induction (without using the PBW theorem). □

It also follows that since $U(\mathfrak{g})$ and $S(\mathfrak{g})$ share the same basis, that they are isomorphic as vector spaces (but not as algebras, since one is commutative while the other is not). Instead of using this basis-dependent isomorphism, we will construct another, basis-independent linear isomorphism $t\colon S(\mathfrak{g}) \to U(\mathfrak{g})$ that is more convenient for our purposes, as follows. Note that there is an isomorphism s between the quotient $S(\mathfrak{g}) = T(\mathfrak{g})/\langle x \otimes y - y \otimes x \rangle$ and the subspace of $T(\mathfrak{g})$ consisting of symmetric tensors, given by

$$X_1 \cdots X_n \mapsto s(X_1 \cdots X_n) = \frac{1}{n!} \sum_{\sigma \in S_n} X_{\sigma(1)} \otimes \cdots \otimes X_{\sigma(n)}.$$

Proposition 3.28. *Denoting $p\colon T(\mathfrak{g}) \to U(\mathfrak{g})$ for the projection of $T(\mathfrak{g})$ to the quotient space $U(\mathfrak{g}) = T(\mathfrak{g})/(x \otimes y - y \otimes x - [x,y])$, the composition $t = p \circ s\colon S(\mathfrak{g}) \to U(\mathfrak{g})$ is a linear isomorphism.*

Once again we point out that this is not an algebra isomorphism.

Proof. Set $S_n(\mathfrak{g}) = \oplus_{i=1}^{n} S^i(\mathfrak{g})$, i.e., all symmetric tensors of order up to n; thus, $S_n(\mathfrak{g})$ is the filtration induced by the gradation of $S(\mathfrak{g})$. We shall prove that t is an isomorphism when restricted to $S_n(\mathfrak{g})$ for each n. The isomorphism maps a $P \in S(\mathfrak{g})$ to $s(P) + K$, where K is the ideal generated by $x \otimes y - y \otimes x - [x,y]$. But if $P \in S(\mathfrak{g})$ is homogeneous then $s(P) \in T(\mathfrak{g})$ is also homogeneous, while no homogeneous elements are contained in the ideal K. Therefore, $s(P)$ will never be an element of K, so if $P \neq 0$ then $t(P) \neq 0$. Therefore t is injective, and since $S_n(\mathfrak{g})$ and $U_n(\mathfrak{g})$ have the same dimension t is also surjective. □

Remark 3.29. For later reference, we give a brief description of the inverse $t^{-1}\colon U(\mathfrak{g}) \to S(\mathfrak{g})$ of the isomorphism t. The inverse of an element of the form $g = \gamma_1 \circ \cdots \circ \gamma_n \in U(\mathfrak{g})$ may be calculated as follows. Neighbouring factors $\cdots \circ \gamma_i \circ \gamma_{i+1} \circ \cdots$ may be swapped at will, at the cost of a commutator $\cdots \circ [\gamma_1, \gamma_{i+1}] \circ \cdots$. Note that the resulting term containing this commutator is in $U_{n-1}(\mathfrak{g})$, i.e., its order with respect to the filtration of U is reduced by one. Thus, switching neighbouring factors in this way, we can rewrite g to $g = \frac{1}{n!} \sum_{\sigma \in S_n} \gamma_{\sigma(1)} \circ \cdots \circ \gamma_{\sigma(n)} + r$, where r contains all the terms with the commutators. Then $t^{-1}(g) = \gamma_1 \cdots \gamma_n + t^{-1}(r)$. Proceed by induction.

3.6.3 The Kontsevich star product

Returning to the algebra of smooth functions $C^\infty(\mathfrak{g}^*)$ on $\mathfrak{g}^*$, we want to explicitly describe the Kontsevich star product $\star$ on $C^\infty(\mathfrak{g}^*)[\![\hbar]\!]$. For this purpose it is convenient to first restrict our attention to polynomials in the coordinate functions x^i, instead of arbitrary smooth functions; afterwards we shall extend the result to all of $C^\infty(\mathfrak{g}^*)$. Recalling that x^i is simultaneously a basis vector of the dual $\mathfrak{g}$ of $\mathfrak{g}^*$ (see Remark 3.24 on p. 62), any polynomial in the coordinate functions x^i may be interpreted as an element of $S(\mathfrak{g})$. Therefore, we now examine the algebra $(S(\mathfrak{g})[\![\hbar]\!], \star)$. Let us set $U_\hbar(\mathfrak{g}) = T(\mathfrak{g})[\![\hbar]\!]/(x \otimes y - y \otimes x - \hbar[x,y])$; that is, $U_\hbar(\mathfrak{g})$ is $U(\mathfrak{g})[\![\hbar]\!]$ but with the bracket rescaled to $\hbar[\,,\,]$. First, we observe the following.

Remark 3.30. Let $p, q \in S(\mathfrak{g})$ be two polynomials on $\mathfrak{g}^*$, of degrees n and m respectively. Then $p \star q$ is not just a formal power series in $\hbar$, but actually a polynomial in $\hbar$ (with coefficients in $S(\mathfrak{g})$) – that is, it terminates. Indeed, consider the coefficient of $\hbar^r$ in the formula (3.6) for $p \star q$. This is a linear combination of expressions of the form $B_{\Gamma,\alpha}(p,q)$ with $\Gamma \in G_r$. Now $B_{\Gamma,\alpha}$ is a differential operator of order $2r$, acting on p, q and the coefficient functions $\alpha^{ij} = c_k^{ij} x^k$ of the Poisson bracket on $\mathfrak{g}^*$. These three are, however, all polynomials, so that if the order $2r$ of $B_{\Gamma,\alpha}$ exceeds the combined order of the coefficient functions α^{ij} and p and q, we obtain zero. Thus, the series (3.6) defining $p \star q$ terminates. For this reason, we are allowed to set $\hbar = 1$, obtaining a product $\star_1$ on $S(\mathfrak{g})$.

Theorem 3.31 ([Kon03]). *If $\star$ is the Kontsevich star product induced by the Poisson structure $\{,\}$ on $\mathfrak{g}^*$, then $(S(\mathfrak{g})[\![\hbar]\!], \star)$ is as an algebra canonically isomorphic to $U_\hbar(\mathfrak{g})$.*

Proof. Taking two polynomials $p, q \in S(\mathfrak{g})$ of degrees n and m again, it follows from the remark above that

$$p \star_1 q = pq + r$$

where $r \in S(\mathfrak{g}^*)$ is a polynomial on $\mathfrak{g}^*$ of order less than $n+m$. Turning this equation around gives $pq = p \star_1 q - r$. Then if $\gamma_i \in \mathfrak{g}$ then $\gamma_1 \cdots \gamma_n = \gamma_1 \star_1 \cdots \star_1 \gamma_n + r$ where r is again a polynomial of order less than n, and it follows by induction that any polynomial may be written as a linear combination of terms of the form $\gamma_1 \star_1 \cdots \star_1 \gamma_n$ for $\gamma_i \in \mathfrak{g}$.

Now consider the algebra $(S(\mathfrak{g}), \star_1)$. Then, the map that sends $\gamma \in \mathfrak{g}$ to $\gamma \in S(\mathfrak{g})$, induces a homomorphism $I : U(\mathfrak{g}) \to S(\mathfrak{g})$ by the universal property of $U(\mathfrak{g})$, i.e., $I(\gamma \circ \eta) = I(\gamma) \star_1 (\eta)$. This map is surjective by the following argument: take an arbitrary polynomial from $S(\mathfrak{g})$, and write it as a linear combination of terms of the form $\gamma_1 \star_1 \cdots \star_1 \gamma_n$, then the same expression but with $\star_1$ replaced by $\circ$ gets mapped precisely onto (the rewritten version of) our polynomial.

Now let $\gamma_1, \gamma_2 \in \mathfrak{g} \subset C^\infty(\mathfrak{g}^*)$. Working out their star product directly using equation (3.17), we get

$$\gamma_1 \star_\hbar \gamma_2 = \gamma_1\gamma_2 + \hbar\alpha^{ij}\partial_i(\gamma_1)\partial_j(\gamma_2) = \gamma_1\gamma_2 + \tfrac{1}{2}\hbar[\gamma_1, \gamma_2].$$

Here, the product $\gamma_1\gamma_2$ is the product in $S(\mathfrak{g})$. But then

$$\gamma_1 \star_\hbar \gamma_2 - \gamma_2 \star_\hbar \gamma_1 = \hbar[\gamma_1, \gamma_2],$$

i.e., on elements of $\mathfrak{g}$, $U_\hbar(\mathfrak{g})$ and $(S(\mathfrak{g})[\![\hbar]\!], \star_\hbar)$ have the same relation. This means that I restricts onto the filtrations $I_n \colon U_n(\mathfrak{g}) \to S_n(\mathfrak{g}) := \oplus_{i=0}^n S^i(\mathfrak{g})$, which are finite dimensional. As these restrictions I_n are surjective by the previous argument, and as the dimensions of $U_n(\mathfrak{g})$ and $S_n(\mathfrak{g})$ agree, this restriction I_n of I is injective for each n. Therefore I itself is injective so that I is indeed an isomorphism.

We can now reinstate $\hbar$ by replacing $S(\mathfrak{g})$ with $S(\mathfrak{g})[\![\hbar]\!]$, $U(\mathfrak{g})$ with $U_\hbar(\mathfrak{g})$ and $[,]$ with $\hbar[,]$. Then I is still an isomorphism. □

This isomorphism yields the formula

$$p \star q = I(I^{-1}(p) \circ I^{-1}(q)),$$

but this does not yet tell us how to compute the star product of two arbitrary polynomials, because we do not know what the inverse of I looks like. This is because the isomorphism I depends on the star product, of which we have not yet found an explicit description. However, working in the $\hbar = 1$ theory again, it is easy to see the following fact.

Proposition 3.32. *The linear isomorphism $t \colon S(\mathfrak{g}) \to U(\mathfrak{g})$, defined in Proposition 3.28, is a homomorphism with respect to $\star$ if and only if t is the inverse of $I \colon U(\mathfrak{g}) \to S(\mathfrak{g})$.*

Proof. Suppose that t and I are inverses of each other, then for $\gamma_i \in \mathfrak{g}$,

$$\gamma_1 \circ \cdots \circ \gamma_n = t \circ I(\gamma_1 \circ \cdots \circ \gamma_n) = t(\gamma_1 \star \cdots \star \gamma_n).$$

If, on the other hand, t is a homomorphism with respect to $\star$ then

$$t \circ I(\gamma_1 \circ \cdots \circ \gamma_n) = t(\gamma_1 \star \cdots \star \gamma_n) = \gamma_1 \circ \cdots \circ \gamma_n. \qquad \square$$

Let us set, for $p, q \in S(\mathfrak{g})$,

$$p \star_G q = t^{-1}(t(p) \circ t(q)); \tag{3.19}$$

this associative star product is called the *Gutt product* [Gut83]. The proposition above then implies that when the map t is a homomorphism with respect to the Kontsevich product $\star$, that $\star$ and $\star_G$ coincide. Since we *do* know the inverse of t (see Remark 3.29), this allows us to calculate $p \star_G q$ for any p, q. When $\mathfrak{g}$ is nilpotent both statements of Proposition 3.32 hold [Kat00], so that the Gutt and Kontsevich star products indeed coincide. When $\mathfrak{g}$ is not nilpotent, then the Gutt star product is equivalent to Kontsevich's one by a gauge transformation [Dit99], so that we may as well identify the two. For example, with the help of Remark 3.29 let us calculate, for $\gamma, \eta \in \mathfrak{g}$ (and reinstating $\hbar$)

$$\begin{aligned}
\gamma^2 \star \eta &= t^{-1}(t(\gamma^2) \circ t(\eta)) = t^{-1}(\gamma \circ \gamma \circ \eta) = \tfrac{1}{3} t^{-1}(3\gamma \circ \gamma \circ \eta) \\
&= \tfrac{1}{3} t^{-1}(\gamma \circ \gamma \circ \eta + 2\gamma \circ \eta \circ \gamma + 2\hbar\gamma \circ [\gamma, \eta]) \\
&= \tfrac{1}{3} t^{-1}(\gamma \circ \gamma \circ \eta + \gamma \circ \eta \circ \gamma + \eta \circ \gamma \circ \gamma + 2\hbar\gamma \circ [\gamma, \eta] + \hbar[\gamma, \eta] \circ \gamma) \\
&= \gamma^2\eta + \tfrac{2\hbar}{3}\gamma[\gamma, \eta] + \tfrac{\hbar}{6} t^{-1}(\gamma \circ [\gamma, \eta] + [\gamma, \eta] \circ \gamma + \hbar[\gamma, [\gamma, \eta]) \\
&= \gamma^2\eta + \hbar\gamma[\gamma, \eta] + \tfrac{\hbar^2}{6}[\gamma, [\gamma, \eta]].
\end{aligned} \tag{3.20}$$

From this formula one may calculate $\gamma_1\gamma_2 \star \eta$ by the identity $\gamma_1\gamma_2 = \frac{1}{4}((\gamma_1 + \gamma_2)^2 - (\gamma_1 - \gamma_2)^2)$. More generally, we have the following formula [Kat00].

Theorem 3.33. *For any $\gamma, \eta \in \mathfrak{g}$ the product $\star$ satisfies*

$$\gamma^n \star \eta = \sum_{k=0}^{n} \hbar^k \binom{n}{k} \widehat{B}_k \, \gamma^{n-k} \mathrm{ad}_\gamma^k(\eta),$$

where $\widehat{B}_k$ is the kth Bernoulli number. This formula determines $\star$ completely.

For example, we can now easily calculate

$$\begin{aligned}
&\gamma^3 \star \eta = \gamma^3\eta + \tfrac{3\hbar}{2}\gamma^2[\gamma, \eta] + \tfrac{\hbar^2}{2}\gamma[\gamma, [\gamma, \eta]] + \tfrac{\hbar^3}{30}[\gamma, [\gamma, [\gamma, \eta]]], \\
&\gamma^k = \gamma^{\star k} \quad \text{for any } k,
\end{aligned}$$

and

$$\begin{aligned}\gamma^2 \star \gamma\eta &= \gamma \star \gamma \star \gamma\eta \\ &= \gamma \star \gamma \star (\gamma \star \eta - \tfrac{\hbar}{2}[\gamma, \eta]) \\ &= \gamma^3 \star \eta - \tfrac{\hbar}{2}\gamma^2 \star [\gamma, \eta].\end{aligned}$$

Example 3.34. Take $\mathfrak{g} = \mathfrak{su}(2, \mathbb{R})$. As a vector space this is just $\mathbb{R}^3$, and the structure constants are ϵ_{ij}^k, the completely antisymmetric tensor. Therefore, the bracket satisfies $[v, w] = v \times w$, where $\times$ is the cross product coming from $\mathbb{R}^3$. This results in the star product discussed in Example 3.2 on p. 47; note that equation (3.20) precisely coincides with (3.2).

This star product on $S(\mathfrak{g})[\![\hbar]\!]$ uniquely extends to $C^\infty(\mathfrak{g}^*)[\![\hbar]\!]$. Indeed, if we know $\star$ on all polynomials, then we may calculate B_n for any n by calculating its coefficient functions and reconstructing B_n from this. Thus, all B_n are fixed and known so that we can calculate $f \star g$ for any two functions $f, g \in C^\infty(\mathfrak{g}^*)[\![\hbar]\!]$.

Chapter 4

How not to deform quantize on jet spaces

At the beginning of the previous chapter, we gave a brief explanation of the classical Hamiltonian formalism on Poisson manifolds. While this formalism can describe a wide range of physical systems, it can essentially handle only point particles. On the other hand, many important physical theories certainly involve more than point particles; a string, for example, is not zero but one-dimensional, and the electromagnetic field is not a point at some location in space, but instead a field that is defined on all of space. Mathematically, these systems are described by partial differential equations, and in particular by a generalization of the Hamiltonian formalism to jet bundles; thus we return our attention to jet bundles. What one would like to do, then, is to generalize Kontsevich's star product to the Hamiltonian formalism on jet space. In this chapter we show that a certain naive way of attempting to do this does not work.

4.1 The variational Hamiltonian formalism

The Hamiltonian formalism on Poisson manifolds that was described briefly at the start of Chapter 3 has a variational generalization, in the sense of secondary calculus (see section 2.2 on p. 26). Before we can describe the variational case, however, we need to slightly reformulate the non-variational case.

Recall that the solution of a Hamiltonian system on a Poisson manifold M is given by the ordinary differential equation

$$\dot{\gamma}(t) = X_H(\gamma(t));$$

here $H \in C^\infty(M)$ is the Hamiltonian, and X_H is the vector field given by $\{\cdot, H\} = \alpha(\mathrm{d}\cdot, \mathrm{d}H)$. In order to make contact with the variational generalization of this, let us write this in a different but equivalent way. The Poisson bivector α induces a fiberwise morphism from T^*M to TM by $\omega \mapsto A(\omega) := \alpha(\cdot, \omega)$, with the understanding that $A(\omega)$ acts on functions f by $A(\omega)(f) = \alpha(\mathrm{d}f, \omega)$. (Note that this morphism is non-invertible unless α is invertible; that is, when M is symplectic.) Then, in coordinates it is easy to see that $A(\omega)(f) = \alpha^{ij}\omega_j\partial_i(f)$, so that

$$A(\omega) = \alpha^{ij}\omega_j \frac{\partial}{\partial x^i},$$

and then, writing (as usual) $\langle\, , \rangle$ for the coupling between differential forms and vector fields,

$$\langle \mathrm{d}f, A(\mathrm{d}g)\rangle = \langle \partial_i(f)\mathrm{d}x^i, \alpha^{kj}\partial_j(g)\partial_k\rangle = \alpha^{ij}\partial_i(f)\partial_j(g) = \alpha(\mathrm{d}f, \mathrm{d}g) = \{f, g\}.$$

Thus, instead of specifying a bivector α, in order to describe a Poisson bracket one may also give a fiberwise morphism $A\colon T^*M \to TM$ satisfying the following two demands:

- For all one-forms ρ, ω we must have $\langle \rho, A(\omega)\rangle = -\langle \omega, A(\rho)\rangle$ so that the resulting bracket $\{f, g\}_A := \langle \mathrm{d}f, A(\mathrm{d}g)\rangle$ is antisymmetric. In coordinates, if $A(\omega_i\,\mathrm{d}x^i) = \alpha^{ji}\omega_i\partial_j$ then this means that $\alpha^{ji} = -\alpha^{ij}$.
- The bracket $\{f, g\}_A = \langle \mathrm{d}f, A(\mathrm{d}g)\rangle$ satisfies the Jacobi identity. This implies the usual PDE on the components α^{ij}, namely that if P is the bivector given by $P(\rho, \omega) = \langle \rho, A(\omega)\rangle$ then $[\![P, P]\!] = 0$ (where $[\![\, ,]\!]$ is the Schouten bracket on $\bigwedge^\bullet TM$).

The equation $\dot\gamma(t) = X_H(\gamma(t))$ specifying the solutions of the system may then be witten as

$$\dot\gamma(t) = A(\mathrm{d}H)|_{\gamma(t)} \tag{4.1}$$

Now let $\pi\colon E \to M$ be a bundle and $J^\infty(\pi)$ be its infinite jet space. Any two-vector $P \in \mathcal{C}\mathrm{Diff}_2(\widehat{\varkappa}(\pi), \overline{H}^n(\pi))$ may be written as $P(b) = \int \langle b, A(b)\rangle$ where $A\colon \widehat{\varkappa}(\pi) \to \varkappa(\pi)$ is an operator in total derivatives, and A satisfies $\int \langle p_1, A(p_2)\rangle = -\int \langle p_2, A(p_1)\rangle$ because P is antisymmetric. If $[\![P, P]\!] = 0$, then P is called a *Poisson bivector*, while A is called a *Hamiltonian operator* (although a Poisson operator would perhaps be a better name). Then it defines a Poisson bracket on the space of functionals $\overline{H}^n(\pi)$ by

$$\{H_1, H_2\}_A = \int \langle \delta H_1, A(\delta H_2)\rangle = P(\delta H_1, \delta H_2)$$

for $H_1, H_2 \in \overline{H}^n(\pi)$. (Sometimes we will also write $\{H_1, H_2\}_P = \{H_1, H_2\}_A$.)

Definition 4.1. When a PDE can be written in the form

$$u_t = A(\delta H)$$

for a certain Hamiltonian operator A and functional $H \in \overline{H}^n(\pi)$, called the *Hamiltonian*,

then the PDE is said to be a *Hamiltonian equation*.

For more details on these matters, see e.g., [KV99, Ch. 5.2, 5.4].

Now that we have a natural generalization of the Hamiltonian formalism and of Poisson brackets to jet bundles, we want to consider possible generalizations of star products. It is, however, not immediately clear on which ring of "functions" such a star product should be defined. Indeed, when the base space M of the jet bundle is $\{\mathrm{pt}\}$, both $\mathcal{F}(\pi)$ and $\overline{H}^n(\pi)$ reduce to $C^\infty(E)$, where E is the total space of the bundle $E \xrightarrow{\pi} \{\mathrm{pt}\}$.

Let us first consider the second option. We remark that considering integral functionals is in perfect agreement with the ideology of secondary calculus (see Section 2.2 on p. 26), where one views sections $s \in \Gamma(\pi_{\mathrm{BV}})$ as "points" and integral functionals from $\overline{H}^n(\pi)$ as particular examples of $\mathbb{R}$-valued "functions" defined at every "point". In order to obtain a well-defined multiplication for functionals $F \in \overline{H}^n(\pi)$, however, we first have to extend $\overline{H}^n(\pi)$ to the larger space defined in equation (2.9) in Ch. 2:

$$\mathfrak{M}(\pi) = \bigoplus_{i=1}^{+\infty} \overline{H}^n(\pi)^{\otimes i}. \tag{4.2}$$

It is clear that this space (as opposed to $\overline{H}^n(\pi)$) is closed under the product $F \cdot G := F \otimes G$ of two functionals F, G. Having a Poisson bivector $P = \int \langle b, A(b) \rangle$, we want the first-order term in $\hbar$ to be the Poisson bracket $\{\,,\}_P$ on $\overline{H}^n(\pi)$. Thus, let us call P *deformable* when the product $\cdot$ on the space $\mathfrak{M}(\pi)$ can be deformed to an associative (but generally non-commutative) star product $\star$ on $\overline{H}^n(\pi)[\![\hbar]\!]$, which for $F, G \in \overline{H}^n(\pi)[\![\hbar]\!]$ is of the form

$$F \star G = F \cdot G + \hbar\{F, G\}_P + O(\hbar)^2.$$

Then the question (stated as Conjecture 9.4 in [Kis12c, Ch. 9]) is the following:

Question 4.2. *Are all bivectors P deformable in this sense?*

We note that the first-order term $\{F, G\}_P$ in this product is an integral functional, while the zeroth order term is not.

Notice, however, that under the evaluation at a section $s \in \Gamma(\pi)$, the functional F becomes an integral $F(s) = \int f(j^\infty_x(s))\, \mathrm{d}^n x$ over the spacetime M; that is, $F(s)$ depends on the values of s at all points of M. From a physical point of view this means that the functional F is a nonlocal operator. On the other hand, consider now a function $f \in \mathcal{F}(\pi)$ on jet space. Under the evaluation on a section $s \in \Gamma(\pi)$, the function f corresponds to a nonlinear differential operator on the components of s; that is, a local operator. Moreover, $\mathcal{F}(\pi)$ carries an obvious multiplication structure, namely the pointwise product of two functions (which is still local). However, it is not immediately clear what Poisson bracket there is on this space which would be the first-order term in $\hbar$ of our star product. Having said that, when M has a nowhere vanishing volume form (coming for example from a Riemannian metric) that we keep denoting by $\mathrm{d}^n x$, then the

Poisson bivector

$$P(F,G) = \int \frac{\delta f}{\delta q^i} A^{ij}_\sigma D_\sigma \left(\frac{\delta g}{\delta q^j} \right) \mathrm{d}^n x \tag{4.3}$$

(where f and g are the densities of F and G, respectively) induces a bracket $\{\,,\}$ on $\mathcal{F}(\pi)$ by simply taking the (antisymmetrized[1]) integrand of the right hand side above:

$$\{f,g\} = \frac{1}{2} \frac{\delta f}{\delta q^i} A^{ij}_\sigma D_\sigma \left(\frac{\delta g}{\delta q^j} \right) - (f \leftrightarrow g).$$

This bracket $\{\,,\}$ is then such that $P(F,G) = \int \{f,g\} \, \mathrm{d}^n x$. Note that it is skew-symmetric by definition, but it need not necessarily satisfy the Jacobi identity. Instead, the Jacobi identity for P implies that $\{\{f,g\},h\} + \circlearrowleft$ is $\overline{\mathrm{d}}$-exact. Now let us again call such brackets deformable when there is an associative star product $\star$ on $\mathcal{F}(\pi)[\![\hbar]\!]$, which for $f, g \in \mathcal{F}(\pi)$ takes the form

$$f \star g = fg + \hbar\{f,g\} + O(\hbar^2).$$

Question 4.3. *Are all brackets $\{\,,\}$ on the ring $\mathcal{F}(\pi)$ such that the associated bracket $\int \{\,,\} \, \mathrm{d}^n x$ on $\overline{H}^n(\pi)$ is Poisson deformable?*[2]

4.2 Three candidate star products

We have seen in Example 3.10 on p. 53 that when the coefficients α^{ij} of the Poisson bivector α on the smooth manifold are constant, then the Kontsevich product reduces to the Moyal product (c.f. Example 3.8 on p. 50). Taking this as a starting point, we might first want to attempt to generalize the Moyal product to jet space. As a naive approach for doing this, we simply take formula (3.4) for the Moyal product on a manifold, and replace every partial derivative ∂_i with a variational derivative $\delta/\delta q^i$. Note that contrary to derivatives ∂_i on smooth manifolds, the variational derivatives $\delta/\delta q^i$ do not commute when acting on some function f, so that different orderings of the variational derivatives yield different star products. In this section we shall consider three possible generalizations of the Moyal product that differ in the ordering of their derivatives, and show that all three are not associative.

Since (for each star product) we can show non-associativity by a single counterexample, we are free to take the following, rather specialized, case. Let $P = \int \langle b, A(b) \rangle = \int b_i A^{ij}_\sigma b_{j,\sigma} \, \mathrm{d}^n x$ be a Poisson bivector. Keeping the Moyal case in mind, we require that $A^{ij}_\sigma = 0$ when $\sigma \neq \emptyset$, and that $A^{ij}_\emptyset = A^{ij}$ is constant. Without loss of generality we may assume that the remaining coefficients are antisymmetric, $A^{ij} = -A^{ji}$. If $F = \int f \, \mathrm{d}^n x$

[1] The fact that the bracket $P(F,G)$ on $\overline{H}^n(\pi)$ is antisymmetric implies only that the integrand of (4.3) is antisymmetric up to trivial terms. Therefore, in order to end up with an antisymmetric bracket on $\mathcal{F}(\pi)$ it is necessary to remove such trivial terms by explicitly taking the antisymmetrization.

[2] This is essentially Conjecture 9.3 from [Kis12c, Ch. 9], modified to take into account the bracket $\{\,,\}$.

and $G = \int g \, \mathrm{d}^n x$ are two functionals, then the bracket $\{\,,\}$ on $\mathcal{F}(\pi)$ which is such that $P(F,G) = \int \{f,g\} \, \mathrm{d}x$ is

$$\{f,g\} = A^{ij} \frac{\delta f}{\delta q^i} \frac{\delta g}{\delta q^j}.$$

For notational ease we henceforth write $\delta_i = \delta/\delta q^i$ and $\partial_i^\sigma = \partial/\partial q^i_\sigma$. Then we postulate the following formula as a first candidate for the term of order n in the candidate star product:

$$B_n^1(f,g) = \frac{1}{n!} A^{i_1 j_1} \cdots A^{i_n j_n} \delta_{i_1} \cdots \delta_{i_n}(f) \delta_{j_1} \cdots \delta_{j_n}(g),$$

which is what one obtains after replacing the ordinary derivatives with variational ones in the n-th order term in the (upper line of) equation (3.4) on p. 51.

Next, we define an operator $E(f,(i_1,\dots,i_n))$ by moving all total derivatives that occur in the expression $\delta_{i_1} \cdots \delta_{i_n}(f)$ to the far left:

$$E(f,(i_1,\dots,i_n)) := (-)^{\sigma_1 \cup \cdots \cup \sigma_n} D_{\sigma_1} \dots D_{\sigma_n} \partial_{i_1}^{\sigma_1} \dots \partial_{i_n}^{\sigma_n}(f). \tag{4.4}$$

Our next candidate for the n-th order term is then

$$B_n^2(f,g) = \frac{1}{n!} A^{i_1 j_1} \cdots A^{i_n j_n} E(f,(i_1,\dots,i_n)) E(g,(j_1,\dots,j_n)).$$

Finally, let us specialize even further to a two-dimensional fiber with coordinates q^1, q^2 and a one-dimensional base with coordinate x. Then the multi-index σ in a coordinate q^i_σ consists only of the number of times a total derivative with respect to x is taken; that is, $q^i_j = D_x^j q^i$. We take the coefficients A^{ij} to be antidiagonal:

$$A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

The third candidate formula is a generalization of equation (3.5) on p. 51:

$$B_n^3(f,g) = \frac{1}{n!} \sum_{i=0}^{n} (-)^i \binom{n}{i} \delta_1^{n-i} \delta_2^i(f) \delta_1^i \delta_2^{n-i}(g).$$

The three candidate star products are then

$$f \star_i g = fg + \sum_{n>0} \hbar^n B_n^i(f,g).$$

Theorem 4.4. *None of the products $\star_i$ for $i = 1,2,3$ are associative.*

Proof. We show the non-associativity by using the following functions:

$$f = h = q^1 q^2, \qquad g = q^1_x q^2_x.$$

Using the Mathematica program that is described and included in the next section, we have calculated the associators $[f,g,h]_i := (f \star_i g) \star_i h - f \star_i (g \star_i h)$ for $i = 1,2,3$. It turns out that for the functions f, g, h defined above the three associators are all equal to

$$[f,g,h]_i = \hbar \left(2(q^1_1)^2 q^2_2 - 2q^2_1 (q^2_1)^2\right).$$

As this is nonzero, all three products are not associative. □

Remark 4.5. Let us briefly attempt to give an intuitive, non-rigorous reason as to why our approach did not work. We believe that the source of the trouble is similar to the reason (sketched in Remark 2.21 on p. 43) why the Laplacian in Section 2.6 did not form a BV-algebra with the Schouten bracket. Indeed, an obvious question here, to which we do not have an answer, is the following: if we take the defining formula (3.4) for the Moyal product and replace all partial derivatives with variational ones, in which order must we place these variational derivatives? The derivatives in the Moyal product (and in the Kontsevich product as well) commute, but the variational derivatives here do not. In the theorem above we have examined three possible answers to this question, none of which resulted in an associative star product.

We have seen in Section 2.6 that composing variational derivatives results in operators that are insensitive to derivative coordinates. Indeed, in Remark 2.21 we explained that we suspect that this is the source of the problems we had in section 2.6. But composing variational derivatives is exactly what we have been doing here, and so it is perhaps not surprising that we again run into problems. We suspect, then, that what we need is the same as what we needed in Section 2.6: a generalization or re-interpretation of the variational derivative that, like functional derivatives, brings with it its own geometry, so that when applied twice these geometries do not interfere with each other.

4.3 The Mathematica program

The program is listed from page 77 to 79. It is also available at `https://sietseringers.net/files/starjet.nb`, and when reading this document on a computer it may also be saved to disk by clicking here.[3]

In order to increase the readability of the program below, let us first briefly comment on some of the features of the Mathematica language.

- Functions take their arguments using square brackets, as in `F[1, 1, 2]`.
- Multiple statements are separated from each other by a semicolon.

[3]Mac OS X users may have to open this document in Adobe Reader, as the Preview app built into OS X seems to be unable to handle PDF attachments.

- Ordered arrays, or lists, are entered by curly braces; thus $\{1, 1, 2\}$ is an example of a list. The elements of such a list need not be just numbers; and not all elements of a list need to be of the same type.
- The operator `@@` is such that when one enters `F@@{1,1,2}`, then `F[1,1,2]` is executed.
- Entering `x // F` is the same as `F[x]`.
- When `x /. y -> z` is entered, then Mathematica replaces all occurrences of `y` in the expression `x` by `z`.
- A new function `F` taking one argument is defined as follows: `F[x_] := ...` The `:=` operator is such that what stands to its right is not evaluated immediately (that is, when Mathematica encounters the function definition of `F`), but only when the newly defined function `F` is executed later on.
- New functions may be defined conditionally, in the following sense. One writes `F[x_] /; expr := ...` The variable `x` may occur in `expr`, which should be such that it evaluates to either `true` or `false`. This particular function definition for `F` is then only used when it is called with an argument `x` that is such that `expr` evaluates to `true`. As an example, if one writes `F[x] /; x==0 := Foo; F[x] /; x!=0 := Bar;` then `F[0]` will return `Foo` while `F[1]` will return `Bar`.
- The (predefined) function `Prepend` takes two arguments, of which the first is assumed to be a list. It returns the same list with its second argument prepended to the first argument. For example, `Prepend[{a,b,c},d]` returns the list `{d,a,b,c}`.
- The (predefined) function `D`$[f, x]$ returns the partial derivative of f with respect to x.
- If Mathematica encounters a symbol (such as $\hbar$) which has not yet been defined, and it does not precede a left square bracket `[` that would indicate that it is a function, then it assumes it to be a (complex) number.

The final point above means that we can achieve linearity in $\hbar$ for the operators $B_n^{j=1,2,3}$ by simply not defining $\hbar$. Finally, in Mathematica a superscript is usually interpreted as a power, so we represent the jet coordinates by $q_{i,j}$; here i is the fiber-index and takes values $i = 1, 2$, while j is the number of derivatives.

Next, we describe the custom functions defined below.

`degree`$[f]$ takes a function f, and returns the differential degree of the function (that is, the smallest $j \in \mathbb{N}$ such that $f \in C^\infty(J^j(\pi))$). It does this simply by compiling a list of all variables $q_{i,j}$ that occur in f, and returning the highest j that it finds. (This means that if the function f is such that its full expression is not known at the time when it is fed to `degree`, then this function will not work . This is not a problem in our case.)

`TD`$[f, n]$ returns the total derivative of f with respect to the base coordinate x; it is implemented directly using equation (1.1). The optional argument n specifies how many times the total derivative of f should be taken. If absent it will assumed to be 1.

`e`$[f, \{i_1, \ldots, i_n\}]$ corresponds to the operator E defined in (4.4). The i_k are fiber-indices, meaning that they should be 1 or 2. Notice that when E (and therefore also `e`)

is given a multi-index that contains a single element, then it coincides with the variational derivative.

VarDer$[f, \{i_1, \ldots, i_n\}]$ returns the iterated variational derivative of f with respect to q^{i_n}, $\ldots$, q^{i_1}. It is implemented using the function `e` as $e[\ldots e[f, i_n], i_{n-1}] \ldots], i_1]$.

star$[B, f, g]$ calculates $fg + \sum_{n=1}^{\texttt{order}} \hbar^n B_n(f, g)$. Here the B in B_n is the first argument of `star`, in order to allow for the calculation of star products associated to multiple sets of differential operators $(B_n)_n$. Since the output is meant to equal the star product only up to the order specified by the `order` variable, terms of order higher than that are truncated.

Assuming the degree of a function f can be determined reliably by `degree` (see the remark in its description above), the functions `TD`, `e` and `VarDer` should always return the correct value for any f of any degree. However, the output of `star` is truncated (at the order specified by the `order` variable), because it is impossible to have a program such as this calculate the full star product as it would contain a sum over an infinite number of terms. On the other hand, we will only use this program to calculate the star product of the functions f, g and h, all of which contain two fiber-coordinates q^i. As all operators $B_n^{j=1,2,3}(f, g)$ defined in the previous section take at least n derivatives with respect to the fiber coordinates, each term of the output of `star` will contain at most two fiber-coordinates q. Therefore, even if `star` would not truncate its output at some order, everything would at most be of order 2 anyway.

Definitions

```
degree[f_] /; Not[NumericQ[f]] :=
  Max[Cases[{f}, q_,_, Infinity] /. {q_i_,j_ → j}];
degree[f_] /; NumericQ[f] := 0;
TD[f_] :=
  D[f, x] + Sum[q1,i+1 D[f, q1,i] + q2,i+1 D[f, q2,i],
    {i, 0, degree[f]}];
TD[f_, 0] := f;
TD[f_, 1] := TD[f];
TD[f_, n_Integer] /; n > 1 := Nest[TD, f, n];

e[f_, args_List] := Module[{i},
   (* First we build up the expression with placeholder
    functions d and td instead of D and TD,
   otherwise D and TD would evaluate expression with
    summation variables which would result in zero. After
    the expression has been built we replace the
    placeholder functions with the actual ones *)
   Sum @@ Prepend[
       (* Generate the list of summation variables and
        their ranges for the total derivatives *)
       Array[{i[#], 0, degree[f]} &, Length[args]],
       (* This is what we sum over *)
       (-1)^Plus@@Array[i[#]&,Length[args]] td[
         d @@ Prepend[
           (* Generate the list of fiber coords wrt to
            which we take the derivative *)
           Array[qargs[[#]],i[#] &, Length[args]],
           (* This is what we differentiate *)
           f
          ],
         Plus @@ Array[i[#] &, Length[args]]
        ]
      ] /. {d → D, td → TD} // Expand
  ];
VarDer[f_, l_List] /; Length[l] == 1 := e[f, l];
VarDer[f_, l_List] /; Length[l] > 1 :=
  e[VarDer[f, Delete[l, 1]], {l[[1]]}];
```

```
a[1,2] = 1;
a[2,1] = -1;
a[2,2] = 0;
a[1,1] = 0;

(* Helper function used in B1 and B2 below *)
IB[n_][f_, g_] /; n > 0 := Module[{k, l},
  1/n! Sum @@ Prepend[
    Join[Array[{k[#], 1, 2} &, n], Array[{l[#], 1, 2} &, n]],
    d[f, Array[k[#] &, n]] Product[a[k[m], l[m]], {m, n}]
     d[g, Array[l[#] &, n]]
   ]
 ];
B1[n_][f_, g_] := IB[n][f, g] /. d -> VarDer;
B2[0][f_, g_] := f g;
B2[n_][f_, g_] := IB[n][f, g] /. d -> e;
B1[0][f_, g_] := f g;
B3[n_][f_, g_] /; n > 0 := Module[{i},
   1/n! Sum[
     (-1)^i Binomial[n, i]
      d[f, Join[Array[1 &, n - i], Array[2 &, i]]]
      d[g, Join[Array[1 &, i], Array[2 &, n - i]]]
     , {i, 0, n}]
  ] /. d -> VarDer;
B3[0][f_, g_] := f g;


(* The star product itself *)
order = 4;
star[B_, f_, g_] := Collect[Sum[ħ^i B[i][f, g], {i, 0, order}], ħ] /.
  ħ^k_ /; k > order :> 0;
Associator[B_, f_, g_, h_] :=
 Collect[
  Expand[star[B, star[B, f, g], h] - star[B, f, star[B, g, h]]],
  ħ];
```

Calculations

```
f = q1,0 q2,0;
h = f;
g = q1,1 q2,1;
Associator[B1, f, g, h]
```

$$\hbar \left(2\, q_{1,1}^2\, q_{2,0}^2 - 2\, q_{1,0}^2\, q_{2,1}^2\right)$$

```
Associator[B2, f, g, h]
```

$$\hbar \left(2\, q_{1,1}^2\, q_{2,0}^2 - 2\, q_{1,0}^2\, q_{2,1}^2\right)$$

```
Associator[B3, f, g, h]
```

$$\hbar \left(2\, q_{1,1}^2\, q_{2,0}^2 - 2\, q_{1,0}^2\, q_{2,1}^2\right)$$

Part II

Identity Management using Credential Schemes

Chapter 5

Preliminaries

Cryptography is mostly concerned with creating efficient algorithms and protocols for which it is infeasible to violate some security feature. Modern-day computers can perform some kinds of calculations very quickly, while we believe them to be notoriously bad at certain other kinds. Cryptographers often exploit this by creating schemes that are such that any algorithm that is able to violate the security feature of the scheme in question would lead to an efficient algorithm for performing the kind of computations that computers are believed to be bad at. This results in schemes that do what we want them to do, and that do not allow actions or events that we do not want to happen.

In this chapter, we will expand on what we mean with algorithms and cryptographic schemes, as well as on the notions of efficient and infeasible computations. We also introduce basic cryptographic notions such as some often-used cyclic groups, elliptic curves and bilinear pairings, signature schemes, zero-knowledge proofs, and credential schemes themselves. Readers who are already familiar with these topics may safely skip this chapter[1] (perhaps with the exception of the last section on credential schemes).

5.1 Algorithms and efficient computations

5.1.1 Turing machines

In order to abstractly reason about algorithms and what they can and cannot do, we will use an idealized computing device called the *Turing machine* [Tur37]. This is a machine that manipulates a tape that is infinitely long on the right hand side, which consists of cells; each such cell contains a symbol from a certain finite non-empty set Σ called the *alphabet*. The alphabet contains a special symbol $\sqcup$ called the *blank symbol*, and all

[1] Much of the contents of these sections can also be found in *Computational Complexity* by Papadimitriou [Pap94], and Volumes 1 and 2 of *Foundations of Cryptography* by Goldreich [Gol00; Gol04].

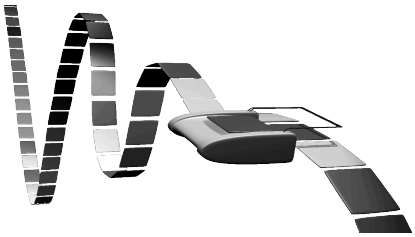


Figure 5.1. Artistic representation of a Turing machine.

but finitely many cells of the tape contain the blank $\sqcup$. The content of the tape between the leftmost cell and the rightmost cell that do not contain $\sqcup$ is called the *input* of the machine.

The computer itself is modeled by a tape-head that is always positioned above a cell. It can read and modify the contents of this cell, and move the tape left or right one cell at a time. The machine is always in a certain state. Denoting the set of all possible states of our machine with K, the machine acts according to a *transition function*:

$$\delta\colon K \times \Sigma \to (K \cup \{\mathsf{halt}, \mathsf{yes}, \mathsf{no}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$$

Given the current state $q \in K$ and the symbol $\sigma \in \Sigma$ currently under the tape-head, this function returns a triple (p, ρ, D), where

- p is the next state of the machine,
- ρ is the symbol to be overwritten on the current position of the tape-head,
- D specifies whether the tape should move left, right, or not at all after having written ρ.

When starting, the machine is always in a special starting state, and the tape-head is positioned above the leftmost cell of the tape. If we denote the left end of the tape as a special symbol $\triangleright$, then we ensure that the tape never falls off the end of the tape by requiring that if q is any state, then $\delta(q, \triangleright) = (p, \triangleright, \rightarrow)$ for some other state p (which may but need not equal q).

We call the three states halt, yes, and no the *ending states*. If p is not one of these three states, then the machine writes the symbol ρ on its current position (overwriting whatever was there previously), and moves left, right, or not at all as specified by D.

If $p = \mathsf{halt}$, then the machine halts and whatever is currently on the tape between the leftmost cell and the rightmost cell that does not contain $\sqcup$ is the *output* of the machine. If $p = \mathsf{yes}$ or $p = \mathsf{no}$ then the machine halts as well, and it is said to have accepted or rejected its input, respectively.

Clumsy though this model may seem at first, it is believed that any algorithm can be expressed in terms of it (and for this reason we shall often use the term "algorithm" and "Turing machine" interchangably). For example, it is easy to reduce a similar machine that operates on multiple tapes simultaneously in each step to the one above. Additionally, we can endow the machine with the ability to sample random data by

giving it access to an extra tape that has been completely filled with random symbols (if a Turing machine has such a tape then we say that it is a *probabilistic* Turing machine; otherwise it is *deterministic*). Finally, if we put one end of a tape into one such machine, and the other end into a second machine, then the output of the first machine can act as input to the second; that is, we now also have a model for *interactive algorithms*.

Sometimes it will be necessary to endow a Turing machine $\mathcal{A}$ with some special capability, in order to study the consequences. We can model this situation as follows. Machine $\mathcal{A}$ is given an extra special tape, of which the other end is connected to an "oracle". $\mathcal{A}$ may write a request on this special tape, that is instantly overwritten by the oracle's answer. For example, the oracle may compute the discrete log of some group element, or give an answer to the Halting Problem for some algorithm-input pair (see Section 5.3). We purposefully do not expand on the nature of the oracle, because how the oracle works is not important; what matters is what happens if machine $\mathcal{A}$ has access to it. This allows us to prove statements of the form "Even if algorithm $\mathcal{A}$ can perform some amount of queries to oracle X, it cannot violate security feature Y."

5.1.2 Computation time and efficiency

If $\mathcal{A}$ is a Turing machine, then an interesting question is the following: given some input x, after how many steps will $\mathcal{A}$ terminate? In fact, $\mathcal{A}$ need not terminate at all: it could be that it moves to the right forever, or always stays at the same position, forever overwriting the cell at that position. When it does halt, however, then we will refer to the number of steps that $\mathcal{A}$ took as the execution time of the machine on input x. If $\mathcal{A}$ does not halt on input x, then we say that its execution time is infinite.

Let $\mathcal{A}$ be a Turing machine (probabilistic or not) operating on alphabet Σ. Denote with Σ^* the set of possible finite concatenations of all symbols from Σ – i.e., all possible contents of the tape of $\mathcal{A}$ (excluding the blank symbols $\sqcup$). Given an input $x \in \Sigma^*$ to machine $\mathcal{A}$, we can define the *length* of $|x|$ as the number of cells that x takes on the tape of M. Now we can define the *time complexity function* $T_{\mathcal{A}} \colon \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ of $\mathcal{A}$ as follows:

$$T_{\mathcal{A}}(n) = \max \big\{ m \in \mathbb{N} \cup \{\infty\} \mid \exists x \in \Sigma^*, |x| = n \\ \text{s.t. } \mathcal{A} \text{ halts after } m \text{ steps when started on input } x \big\}.$$

That is, for all inputs of length n, $T_{\mathcal{A}}(n)$ is the maximum time $\mathcal{A}$ takes to compute its output.

Definition 5.1. A Turing machine $\mathcal{A}$ is said to be *polynomial-time* if there there exists a polynomial $p \colon \mathbb{N} \to \mathbb{N}$ and an integer n_0 such that $T_{\mathcal{A}}(n) \leq p(n)$ whenever $n > n_0$.

Note that if $\mathcal{A}$ is polynomial-time then it must halt on any possible input x.

Much of the literature has adapted the following convention: *Efficient computations are those that can be carried out by probabilistic polynomial-time Turing machines.* One reason for this is that many desirable properties and reductions of Turing machines are stable under this assumption. For example, if a multi-tape Turing machine is polynomial-time, then so is its reduction to a Turing machine that operates on one tape. Accordingly,

if a computation takes superpolynomial time (in the sense that there exists no probabilistic polynomial-time Turing machine that can perform the computation), then it is considered to be infeasible.[2]

An important related notion is that of *negligible functions*. We write $\nu(\ell) < \mathrm{negl}(\ell)$ when the function $\nu\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible*; that is, for any polynomial p there exists an ℓ_p such that $\nu(\ell) < 1/p(\ell)$ if $\ell > \ell_p$. This allows us to define a second notion of infeasibility: if $P(\mathcal{A}(x) \to y)$ denotes the probability that algorithm $\mathcal{A}$ outputs y on input x, and this probability is negligible in $|x|$, then we consider this to be infeasible. This is related to being polynomial-time in the following way: if $P(\mathcal{A}(x) \to y)$ is negligible in $|x|$ for any polynomial-time algorithm $\mathcal{A}$, and algorithm $\mathcal{A}'$ outputs y with a probability that does not depend on $|x|$, then $\mathcal{A}'$ cannot be polynomial-time.

The parameter ℓ is called the *security parameter*.[3] Its purpose is the following. Whenever the running time of an algorithm $\mathcal{A}$ is superpolynomial, we can increase the parameter ℓ until the running time of $\mathcal{A}$ becomes so vast that effectively no existing computer can perform the calculation. Alternatively, if the running time of $\mathcal{A}$ is polynomial but its chance of success is negligible in ℓ, then by increasing ℓ we can make the chance of $\mathcal{A}$ winning so close to zero that it is truly negligible.

5.1.3 Conventions and notations

As above, we will usually denote algorithms and Turing machines with calligraphic letters such as $\mathcal{A}$ and $\mathcal{B}$. If $\mathcal{A}$ is an algorithm, then by $y \leftarrow \mathcal{A}(x)$ we denote that y was obtained by running $\mathcal{A}$ on input x. If $\mathcal{A}$ is a deterministic then y is unique; if $\mathcal{A}$ is probabilistic then y is a random variable. If $\mathcal{A}$ and $\mathcal{B}$ are interactive algorithms, we write $a \leftarrow \mathcal{A}(\cdot) \leftrightarrow \mathcal{B}(\cdot) \to b$ when $\mathcal{A}$ and $\mathcal{B}$ interact and afterwards output a and b, respectively. If algorithm $\mathcal{A}$ has oracle access to an oracle O, we write $\mathcal{A}^O$.

Although the alphabet in terms of which the input to our machines is encoded can in principle be anything, we will henceforth assume that bits $\{0,1\}$ are used for encoding inputs, so that $|x|$ will denote the length of x in bits. For example, if x is a positive integer then $|x| = \lceil \log_2 x \rceil$.

5.2 Groups and group families

Nearly all cryptographic schemes operate in families of finite cyclic groups $\mathbb{G} = \{G_\alpha\}_{\alpha \in I}$, where I is some infinite but countable index set. The groups G_α are generally similar in structure but have increasing order. This allows one to prove that violating a certain

[2] This sense of whether a computation is feasible will not always align with the actual feasibility of a computation. For example, an algorithm that takes k^{100} steps on inputs of length k will cease to be computable very quickly, even though it is polynomial-time. For that reason it is important to try to maximize the efficiency of our schemes, even if they are already polynomial-time. Similarly, if $T_{\mathcal{A}}(k) = k^{100}$ then it need not be the case that the execution time is k^{100} for *all* inputs of length k – perhaps there exists only a single input of that length that takes that long.

Besides these remarks there are many more subtleties and other things to be said about this subject. For a more comprehensive discussion about these matters we refer to [Pap94].

[3] Many texts use the letter k for the security parameter, but we will reserve that letter to denote the attributes of attribute-based credential schemes in later chapters.

security feature becomes increasingly difficult as one moves to larger groups within the group family.[4]

If $\mathbb{G}$ is such a group family, then an *instance generator* for $\mathbb{G}$ is a probabilistic polynomial-time algorithm that, given an integer ℓ in unary,[5] outputs an $\alpha \in I$ and generator $g \in G_\alpha$. Note that then $|\alpha|$ must be polynomially bounded by ℓ, as no polynomial-time algorithm can write output of superpolynomial length. In particular, this implies that if a function is negligible in ℓ, then it is also negligible in $|\alpha|$.

Let us now describe a number of group families (and how they can be generated) that are commonly used in cryptographic schemes. Only the last of these will occur in the next chapters of this thesis. Below and henceforth, we write $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z}$ for the ring of integers modulo n.

Example 5.2 (Quadratic residues). Given 1^ℓ, choose a prime p such that $p' = (p-1)/2$ is also prime with $|p'| = \ell$. Then the group of *quadratic residues* $QR_p \subset \mathbb{Z}_p^*$ is the subgroup of $\mathbb{Z}_p^*$ consisting of the elements that have square roots – i.e., $x \in QR_p$ only if there is an $y \in \mathbb{Z}_p$ such that $x = y^2 \bmod p$; or, more concisely,

$$QR_p = (\mathbb{Z}_p^*)^2 = \{y^2 \mid y \in (\mathbb{Z}/p\mathbb{Z})^*\}.$$

It is easy to see that this group consists of those elements from $\mathbb{Z}_p^*$ whose discrete logarithm with respect to some fixed generator is even (note that since the order $\phi(p) = p-1$ of $\mathbb{Z}_p^*$ is even, evenness of such a discrete log $a \in \mathbb{Z}_{p-1}$ does not depend on the representative of $a \bmod p-1$; also, one can show that if the mentioned property holds for one generator, then it also holds for any other generator). Since there are $\phi(p)/2 = (p-1)/2 = p'$ such elements, the order of QR_p is the prime p'.

Example 5.3 (RSA groups). Given 1^ℓ, choose two distinct primes p, q such that the lengths of p and q are close to ℓ (these values may "differ in length by a few digits" [RSA78]). Set $n = pq$, and return p, q. Then n is the public key and p, q the private key of the well-known and heavily used RSA signature scheme [RSA78].

Example 5.4 (Quadratic residues in RSA groups). Given 1^ℓ, choose two distinct primes p, q such that the lengths of p and q are close to ℓ as above, and such that $p' = (p-1)/2$ and $q' = (q-1)/2$ are also prime. Set $n = pq$. Then the group of quadratic residues $QR_n \subset \mathbb{Z}_n^*$, defined the same as above, is a cyclic subgroup of order $p'q'$ in $\mathbb{Z}_n^*$. The Idemix credential scheme [CL01] is defined in this group.

[4] We speak of group families instead of just groups mostly in order to have a parameter (in this case $|\alpha|$) in which quantities can be negligible. In future chapters we will often be more sloppy, and say for example "If G is a finite cyclic group of order p then quantity so-and-so is negligible in $|p|$", with the understanding that then G is part of a group family whose members have increasing order.

[5] That is, KeyGen is given a string "$111\cdots111$" of length ℓ (henceforth we will write 1^ℓ for this string). This is done because we want the instance generator to be polynomial-time in ℓ itself, not in the amount of digits that ℓ has in binary or decimal.

Example 5.5 (Elliptic curves). Given 1^ℓ, let q be a prime or prime power, and let $E(\mathbb{F}_q)$ be an elliptic curve over the unique field[6] $\mathbb{F}_q$ of order q that contains a subgroup G of prime order p, where p should be of length ℓ. Return the parameters specifying G and a random generator of G. For example, if q is not a power of 2 or 3, then the curve may be specified by a Weierstrass equation

$$E : y^2 = x^3 + ax + b$$

together with a point $P \in E(\mathbb{F}_q)$ of order p.

Nowadays many cryptographic schemes operate within such groups, such as for example ECDSA, ECDH, the Boneh–Boyen signature scheme (see Chapter 7), and our own attribute-based credential scheme (see Chapter 9). U-Prove [Bra00] can also be implemented in such groups.

We are purposefully rather vague here in how the curve should be selected, because this varies widely depending on the application. We will describe one construction that is suitable for the schemes of Chapters 7 and 9 in Section 5.4 below.

In each of these examples it is common to assume the difficulty of computing discrete logarithms (see Definition 5.6 below). Often the Decisional Diffie-Hellman assumption is also taken (Definition 5.10), although for certain elliptic curves this assumption does not hold (see Section 5.4).

5.2.1 Conventions and notations

We will often encounter elements from the $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z}$ for some integer n. As mentioned above, we will consistently refer to this ring as $\mathbb{Z}_n$, also when n is prime so that $\mathbb{Z}_n$ is a field. If $a \in \mathbb{Z}_n$ then we will usually refer to the equivalence class $a \bmod n \in \mathbb{Z}_n$, as opposed to, for example, the smallest positive element of this equivalence class. This means that if $a, b \in \mathbb{Z}_n$ and we write for example a/b, then we refer to modular division.[7]

Whether the group at hand is written additively or multiplicatively depends on the application, the number-theoretic origin of the group, on the conventions of the literature and on the preference of the author. For example, $\mathbb{Z}_n^*$ is almost always written multiplicatively for obvious reasons. In number-theoretic literature elliptic curves are almost always written additively, but in cryptography multiplicative notation seems to be used more often, also in the case of elliptic curves. The difference between the two is only notational, in the sense that mathematically they are completely equivalent, and so we will sometimes use the one and then the other. We will, however, adopt the following convention: if G is some group (other than $\mathbb{Z}_n$ for some n) then we will always denote elements from G with capital letters such as X and P, while we write coefficients from $\mathbb{Z}_{|G|}$ with lower or greek letters.

[6] Such a field is often also called a *finite field* or *Galois field*, and sometimes denoted with $\mathrm{GF}(q)$. For any prime power, such a field exists and it is unique in the sense that all other fields of the same prime power order are isomorphic to it. When q is prime it is isomorphic to $\mathbb{Z}_q$, but when q is a prime-power then these two are distinct (in fact, in this case $\mathbb{Z}_q$ is not a field).

[7] In Chapter 7 we will deviate from some of these rules.

5.3 Intractability assumptions

Now that we have a model for computers and algorithms and an understanding of what they can efficiently do, it is time to turn to what they can*not* do. Unfortunately, proving that there exists no algorithm that solves a particular problem is often significantly more difficult than showing that there does exist an algorithm solving some problem.[8] Instead one often proves statements of the form "If no algorithm can efficiently solve problem X, then no algorithm can efficiently solve problem Y". For that reason one often *assumes* the computational intractability of solving some well-studied problem, and then shows that it follows from this assumption that a desired security feature holds. The most well-known intractability assumption of this kind is probably the following.

Definition 5.6 (Discrete logarithm (DL) assumption). Let $\mathbb{G} = \{G_\alpha\}_\alpha$ be a family of finite cyclic groups. The Discrete Logarithm assumption holds in $\mathbb{G}$ if no probabilistic polynomial-time algorithm can, when given (α, P, P^k) output k with probability that is non-negligible in $|\alpha|$.

Sometimes we will write our group G additively instead of multiplicatively as above; in that case the problem would ask for k where $K = kP$.[9]

Let G be a finite cyclic group of order p, which we now assume to be prime, and let $P_1, \ldots, P_n \in G$ all be distinct generators. If for some $K \in G$ the numbers $(k_1, \ldots, k_n) \in \mathbb{Z}_p^n$ are such that $K = \prod_{i=1}^n P_i^{k_i}$, then $(k_1, \ldots, k_n)$ is called a *DL-representation* of K with respect to $(P_1, \ldots, P_n)$. When $k_1 = \cdots = k_n = 0$, so that $K = 1$, we say that the DL-representation is a trivial DL-representation of 1. As an example of the kind of reasoning mentioned above, we now show the following.

Proposition 5.7. *Let $\mathbb{G} = \{G_\alpha\}_\alpha$ be a family of finite cyclic groups of prime order, in which the DL-assumption holds. Then no probabilistic polynomial-time algorithm can, on input α and $(P_1, \ldots, P_n)$ where the P_i are randomly generated, create a non-trivial DL-representation of $1 \in G$ with respect to $(P_1, \ldots, P_n)$ with non-negligible probability in $|\alpha|$.*

Proof. (From [Bra00, p. 60]). Suppose that such an algorithm $\mathcal{A}$ does exist. We construct an algorithm $\mathcal{B}$ that, on input $K, L \in G$ computes $\log_K L \bmod p$ using $\mathcal{A}$ as follows:

1. $\mathcal{B}$ generates $2n$ random numbers $r_1, \ldots, r_n, s_1, \ldots, s_n \in \mathbb{Z}_p$, and sets, for each i,

$$P_i = K^{r_i} L^{s_i}.$$

2. $\mathcal{B}$ then executes $(k_1, \ldots, k_n) \leftarrow \mathcal{A}(\alpha, P_1, \ldots, P_n)$, and checks whether the output of $\mathcal{A}$ is indeed a non-trivial DL-representation of 1. If it is not then $\mathcal{B}$ halts.

[8] A notable exception to this is Turing's negative answer to the Halting Problem, which asks if there exists an algorithm that, when given any other algorithm $\mathcal{A}$ and input x, can decide whether $\mathcal{A}(x)$ will halt or not. In the paper where he introduced Turing machines [Tur37], Turing proved that no such Turing machine exists.

[9] Like many assumptions of this kind, its difficulty depends on how we represent the group elements. For example, let G be a finite group of prime order p (for example, G could be a subgroup of an elliptic curve, see Example 5.5). Then the discrete logarithm problem may hold in G while it definitely does not hold in $\mathbb{Z}_p$, even though these groups are isomorphic (indeed, if $K = kP$ for $K, k, P \in \mathbb{Z}_p$ then the algorithm that computes $k = KP^{-1}$ will be very efficient). We will briefly touch on a similar subtlety in Section 9.5.2 on p. 175.

3. If $\sum_{i=1}^{n} s_i k_i = 0 \bmod p$ then $\mathcal{B}$ halts.
4. $\mathcal{B}$ outputs

$$-\left(\sum_{i=1}^{n} r_i k_i\right)\left(\sum_{i=1}^{n} s_i k_i\right)^{-1} \bmod p.$$

Since

$$\begin{aligned} 1 = \prod_{i=1}^{n} P_i^{k_i} &= \exp\left(K, \sum_{i=1}^{n} r_i k_i\right) \exp\left(L, \sum_{i=1}^{n} s_i k_i\right) \\ &= \exp\left(K, \sum_{i=1}^{n} r_i k_i + (\log_K L) \sum_{i=1}^{n} s_i k_i\right) \end{aligned}$$

it follows that the exponent on the right hand side must equal 0:

$$\sum_{i=1}^{n} r_i k_i + (\log_K L) \sum_{i=1}^{n} s_i k_i = 0 \bmod p,$$

so that the output of $\mathcal{B}$ is indeed $\log_K L$.

It remains to show that the chance of success of $\mathcal{B}$ – that is, the chance that $\mathcal{B}$ does not terminate in step 3 – is not negligible. Let ϵ be the chance that $\mathcal{A}$ succeeds. The output $(k_1, \ldots, k_n)$ of $\mathcal{A}$ cannot depend on the numbers s_i, because they are hidden from $\mathcal{A}$ because of the randomness of the numbers r_i. For any tuple $(s_1, \ldots, s_{n-1})$ there exists exactly one number s_n which is such that $\sum_{i=1}^{n} s_i k_i = 0 \bmod p$, so that there are p^{n-1} "bad" tuples among the total of p^n tuples. Since $\mathcal{B}$ chooses the numbers s_i randomly, the change that it chooses a "bad" tuple is therefore $p^{n-1}/p^n = 1/p$, so that there is a chance $1 - 1/p$ that it chooses a "good" tuple.

Since step 4 takes place only if step 3 is successful, and step 3 takes place only if step 2 is successful, the overall success probability of $\mathcal{B}$ is $\epsilon(1 - 1/p)$ which is not negligible if ϵ is not negligible. □

Although they do not play as large a role in this thesis as the discrete logarithm-problem and the above consequence, we also mention the following two assumptions.

Definition 5.8 (Integer factorization). The integer factorization problem states that no probabilistic polynomial-time algorithm exists that can, given a composite $n = pq$ where p, q are two primes, factor n with probability that is non-negligible in $\min(|p|, |q|)$.

Definition 5.9 (Computational Diffie-Hellman assumption (CDH)). Let $\mathbb{G} = \{G_\alpha\}_\alpha$ be a group family. The Computational Diffie-Hellman assumption holds in $\mathbb{G}$ if there exists no probabilistic polynomial-time algorithm $\mathcal{A}$ that can, given an α along with a tuple (P, P^a, P^b) output P^{ab} with probability that is non-negligible in $|\alpha|$.

This assumption is used in the Diffie-Hellman key exchange protocol [DH76], which is one of the cornerstones of the internet. It allows two parties to establish a shared secret

in the presence of eavesdroppers, which they can then use, for example, as the key in a symmetric encryption protocol in order to securely communicate over their insecure channel.

A closely related assumption is the following.

Definition 5.10 (Decisional Diffie-Hellman assumption (DDH)). Let $\mathbb{G} = \{G_\alpha\}_\alpha$ again be a group family. The Decisional Diffie-Hellman assumption holds in $\mathbb{G}$ if there exists no probabilistic polynomial-time algorithm $\mathcal{A}$ that can, given an α along with a tuple (P, P^a, P^b, P^c) tell whether $c = ab$ (i.e., whether the tuple is a valid CDH instance) with probability that is non-negligible in $|\alpha|$. More precisely, for any probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function ν such that

$$\left| \Pr[\mathcal{A}(\alpha, P, P^a, P^b, P^{ab}) = 1] - \Pr[\mathcal{A}(\alpha, P, P^a, P^b, P^c) = 1] \right| = \nu(|\alpha|),$$

where the probability is over the choice of $P \in G_\alpha$, $a, b, c \in_R \mathbb{Z}_p$ (p being the order of G_α), and the random bits used by $\mathcal{A}$.

Notice that the DDH assumption implies the CDH assumption (since if CDH is easy then DDH is too), and both of them imply the discrete logarithm assumption (otherwise we could just take the discrete logs of the latter three elements with respect to P and examine the result). We will rephrase the DDH assumption later in terms of computational indistinguishability (see Example 5.13).

5.4 Elliptic curves and bilinear pairings

Many cryptographic schemes, including the ones from Chapters 7 and 9, are defined in cyclic groups G whose order is prime, such as for example $QR_q \subset \mathbb{Z}_q$ for some prime q, or (subgroups of) elliptic curves. The structure of these groups is fairly simple: for example, each element unequal to 1 is a generator[10], and the set of coefficients ($\mathbb{Z}_p$ if the order of G is p) is a field (so that if $P \in G$ and $k \in \mathbb{Z}_p^*$, then $P^{1/k}$ exists for any P and k).

In such situations there are at least two reasons for choosing elliptic curves over number theoretic groups such as QR_p. The first is that for the latter there exist more efficient methods for computing discrete logarithms than for elliptic curves, so that elliptic curves offer the same security against discrete logarithm-attacks at smaller sizes (see, e.g., [LV01; Len05]). The second advantage is that some elliptic curves admit *bilinear pairings*: efficiently computable bilinear maps that take two group elements and return a third element.

Definition 5.11. A bilinear group pair (G_1, G_2) consists of two cyclic groups, both of prime order p, such that there exists a *bilinear map* or *pairing*; that is, a map $e\colon G_1 \times G_2 \to G_T$ (with G_T another multiplicative group of order p) satisfying the following properties:

[10] To see this, let $P \in G$ be a generator (such a generator always exists; we will not show this here). If Q is any other element different from 1, then it can be written as $Q = P^q$ for some nonzero $q \in \mathbb{Z}_p$. Now if $R = P^r$ is yet another element, then $R = P^r = P^{qr/q} = Q^{r/q}$. The fraction in the power exists because $\mathbb{Z}_p$ is a field. Thus Q is also a generator, and since Q was arbitrary, any element from G unequal to 1 is a generator.

- *Bilinearity*: For all $P, P' \in G_1$ and $Q, Q' \in G_2$ we have $e(PP', Q) = e(P, Q)e(P', Q)$ and $e(P, QQ') = e(P, Q)e(P, Q')$; or equivalently, $e(P^a, Q^b) = e(P, Q)^{ab}$ for any $a, b \in \mathbb{Z}_p$.
- *Non-degeneracy*: If $P \in G_1$, $Q \in G_2$ are two generators, then the element $e(P, Q)$ is a generator of G_T (that is, it is unequal to $1 \in G_T$).
- *Computability*: There exists an efficient algorithm for computing $e(P, Q)$ for any $Q \in G_1$, $Q \in G_2$.

There are three distinct types of pairings:

1. $G_1 = G_2$, or $G_1 \cong G_2$ and both sides of the isomorphism are efficiently computable.[11]
2. There exists an efficiently computable isomorphism from G_2 to G_1 but not vice versa.
3. There exist no efficiently computable isomorphism from G_2 to G_1 or vice versa.

Notice that in group pairs of Type 1, the DDH-problem cannot hold in $G_1 \cong G_2$: if $(P, P^a, P^b, P^c) \in G_1^4$ is given then one can check whether

$$e(P, P^c) \stackrel{?}{=} e(P^a, P^b)$$

holds, which will only be the case if $c = ab$. For a similar reason, the DDH-problem will not hold in G_2 in the case of a Type 2 pairing (but it might hold in G_1).

The importance and utility of pairings in cryptography became clear in 2000 when Joux devised a three-party one-round key agreement protocol that uses Type 1 pairings [Jou00]. In the years after that many other novel Type 1 schemes were created, but in recent years attention has shifted to Type 3 pairings, mostly for the following reasons.

- Type 1 pairings must be instantiated either using curves defined over fields of characteristic 2 or 3 (i.e., $\mathbb{F}_{2^\ell}$ or $\mathbb{F}_{3^\ell}$ for some ℓ), or using a supersingular curve over a prime field. However, recent advances in solving the discrete logarithm problem in fields of characteristic 2 and 3 [Adj+15; GKZ14] have made the former untenable, while the performance of the latter is significantly less than that of Type 3 pairings.[12]
- Type 3 pairings also perform better than Type 2 pairings. Additionally, there is some evidence that the presence of the efficiently computable isomorphism from G_2 to G_1 of Type 2 pairings is not essential, in the sense that most or all Type 2 schemes can be converted to Type 3 schemes [CM11].

For these reasons we will prefer Type 3 pairings in this thesis. We will mostly use them to check that two elements from, say, G_1 have some specific relative discrete logarithm (without knowing this discrete logarithm). This goes as follows. Let P, Q be generators

[11] Of course, the two groups are always isomorphic, the isomorphism being $P^k \mapsto Q^k$ for generators P, Q of G_1, G_2 – but this isomorphism is not efficiently computable if the DL-problem holds.

[12] For an informal discussion on these issues, we refer to this blog post by S. Galbraith: *"New discrete logarithm records, and the death of Type 1 pairings"*, https://ellipticnews.wordpress.com/2014/02/01/new-discrete-logarithm-records-and-the-death-of-type-1-pairings/

of G_1, G_2 respectively, and let $A = Q^a$ be known (but not a). If $P' \in G_1$ is then some other element then we can check whether or not $P' = P^a$ by checking

$$e(P, A) \stackrel{?}{=} e(P', Q).$$

Such pairings exist for particular types of elliptic curves; we mention for example [MNT01] and the BN-curves [BN06], which we describe below. For more information about bilinear group pairs and pairings we refer to [GPS08]; see also, for example, Chapters I and X from [BSS05].

5.4.1 BN-curves

As an example of a group pair that is suitable for Type 3 pairings, we briefly describe the construction of Barreto-Naehrig curves [BN06]. These are curves over $\mathbb{Z}_q$ for a prime q of the form $y^2 = x^3 + b$. They are parameterized by the following two polynomials in the integer u, which may be positive or negative:

$$q = q(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1,$$
$$p = p(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1.$$

A BN-curve is generated as follows.

- Given 1^ℓ, find a u such that both $p(u)$ and $q(u)$ are prime, and such that $|p(u)| = \ell$;
- Find a $b \in \mathbb{Z}_q$ such that $b + 1 = y^2 \bmod q$ for some $y \in \mathbb{Z}_q$, and such that $pP = \infty$, where $P = (1, y)$;
- Return p, q, b, P.

Then $G_1 = E(\mathbb{Z}_q)$ is a curve of order p and $P \in G_1$ is a generator. Actually it suffices to return just u and y, since using u the values of p and q can be computed using the polynomials above, and $b = y^2 - 1 \bmod q$.

The embedding degree of these curves is 12, meaning that the second group G_2 of these group pairs is of the form $E(\mathbb{F}_{q^{12}})[p]$. However, Baretto and Naehrig noticed that for these curves there will exist a $\xi \in \mathbb{F}_{q^2}$ such that there exists a *twist* E' of E, of the form $E' : y^2 = x^3 + b/\xi$ which is such that $p | \#E'(\mathbb{F}_{q^2})$, along with an isomorphism $\psi : E' \to E$. This means that using this twist we can take G_2 to be $G_2 = \psi^{-1}(E(\mathbb{F}_{q^{12}})[p]) = E'(\mathbb{F}_{q^2})[p]$, so that the elements of G_2 are not much bigger than those of G_1. Additionally, ξ can be chosen such that $X^6 - \xi$ is irreducible over $\mathbb{F}_{q^2}$, meaning that $\mathbb{F}_{q^{12}}$ can be represented as $\mathbb{F}_{q^2}[X]/(X^6 - \xi)$, leading to further speedups in the computation of the pairing.

On these curves one can use for example the Tate or Ate pairing. We will not detail the algorithms for these pairings; but see for example [DSD07] as well as the citations in Definition 5.11.

5.5 Zero-knowledge proofs

In many cryptographic schemes it is necessary for one party (the *prover*) to convince another party (the *verifier*) that it knows some secret, without disclosing it. For example, we might want to prevent the verifier from learning some sensitive piece of information for privacy purposes. For this reason we must expand on what it means for an algorithm "to know" something – or more precisely, what an interactive algorithm learns from an interaction with some other interactive algorithm.

Simply put, we shall say that the interactive algorithm has learned nothing new if whatever data or knowledge it obtained from the interaction, it could also have computed by itself beforehand, without any interaction. Specifically, for any verifier there will exist a (noninteractive) polynomial-time algorithm called the *simulator*, whose output will be indistinguishable from everything the verifier learns when interacting with the prover. This captures that the verifier learns nothing from an interaction with the prover, as it could have computed the result of the interaction without the help of the prover in the first place, by invoking this simulator.

Before we can turn to the definition of zero-knowledge proofs we must define what it means for two sets to be computationally indistinguishable, and introduce some notations for interactive algorithms.

5.5.1 Computational indistinguishability

Both the prover and the verifier may be probabilistic protocols, so that their output may vary even if the input does not. For that reason, the notion of computational indistinguishability that we need is defined in terms of random variables.

Definition 5.12 (Computational indistinguishability). Let S be a countable set. A *probability ensemble* indexed by S is a sequence of random variables $\{X_s\}_{s\in S}$ indexed by S.

Two probability ensembles $X = \{X_s\}_{s\in S}$ and $Y = \{Y_s\}_{s\in S}$ are *computationally indistinguishable* (in which case we write $X \overset{c}{\approx} Y$) if for every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function ν such that

$$\Big|\Pr[\mathcal{A}(X_s, s) = 1] - \Pr[\mathcal{A}(Y_s, s) = 1]\Big| = \nu(|s|).$$

Thus as the length of s increases, it becomes progressively more unlikely that $\mathcal{A}$ reacts one way when given a value from X_s but another way when given a value from Y_s.

Example 5.13. The DDH problem (see Definition 5.10) in a group family $\mathbb{G} = \{G_\alpha\}_{\alpha\in I}$ can concisely be formulated in terms of computationally indistinguishability as follows. If

$$X_\alpha = (P, P^a, P^b, P^{ab}) \quad \text{and} \quad Y_\alpha = (P, P^a, P^b, P^c),$$

are two random variables where P is randomly selected from G_α and a, b, c are random integers, then the DDH problem holds in $\mathbb{G}$ if $\{X_\alpha\}_{\alpha\in I} \overset{c}{\approx} \{Y_\alpha\}_{\alpha\in I}$.

5.5.2 Interactive algorithms

Let $\mathcal{A}$ be an interactive algorithm interacting with some other algorithm $\mathcal{B}$. Suppose that both algorithms are given some input x, while $\mathcal{A}$ is additionally given input w and a randomness tape r. The *next-message function* $\mathcal{A}_{x,w,r}$ is a function that, when given a set of messages $(m_1, \ldots, m_t)$, returns what $\mathcal{A}$ would have output after having received the messages $(m_1, \ldots, m_t)$.

Definition 5.14 (Black-box access). We say that $\mathcal{B}$ has *black-box access* to $\mathcal{A}$, and we write $\mathcal{B}^{\blacksquare}\!\!\rightarrow\mathcal{A}$, if $\mathcal{B}$ has oracle access to the next-message function $\mathcal{A}_{x,w,r}$ of $\mathcal{A}$.

Definition 5.15. We denote with $\text{view}_{\mathcal{A}}(\mathcal{A}(x,w) \leftrightarrow \mathcal{B}(x,a))$ a random variable containing x, w, the random tape of $\mathcal{A}$, and the messages that $\mathcal{A}$ receives during a joint conversation with $\mathcal{B}(x,a)$.

5.5.3 Formal languages and zero-knowledgeness

For our purposes it suffices to define a *formal language* L as some subset of all possible sequences of bits, i.e., $L \subset \{0,1\}^*$.

Definition 5.16. Let L be a formal language. A *relation for* L is a subset $R_L \subset \{0,1\}^* \times \{0,1\}^*$ such that

- There exists an algorithm that is polynomial-time in its first argument, that when given $(x,w) \in \{0,1\}^* \times \{0,1\}^*$ can recognize whether or not $(x,w) \in R_L$,
- There exists a polynomial p such that $x \in L$ if and only if there exists some $w \in \{0,1\}^*$ with $|w| \leq p(|x|)$ and $(x,w) \in R_L$.

Such a w is called a *witness for membership*, or just a *witness*, for $x \in L$.

The complexity class NP can then be defined as all formal languages L for which there exists such a relation R_L. That is, given a witness w for x it is easy to determine that $x \in L$, but in the absence of such a witness it may be infeasible to decide if $x \in L$.

Fix a language L with relation R_L, and suppose one party called the prover $\mathcal{P}$ knows a witness w for x. A zero-knowledge proof is, informally, an interactive protocol that the prover $\mathcal{P}$ can perform with some other party (called the *verifier* $\mathcal{V}$), in which the prover uses the witness w for x in order to convince the verifier that $x \in L$. If both parties follow the protocol then by the end the verifier is convinced that indeed $x \in L$, but if $x \notin L$ (i.e., the prover is cheating) then it should not be able to convince the verifier with any reasonable probability (this property is called soundness). Additionally, as stated above, the protocol should be such that the verifier learns nothing that it could not have computed by itself beforehand.

A zero-knowledge proof *of knowledge* is a zero-knowledge proof which is such that by the end of the protocol, the verifier is not only convinced of the fact that $x \in L$, but also that the prover knows the witness w which is such that $(x,w) \in R_L$. We will formalize this by demanding that if the verifier can arbitrarily manipulate (i.e., has black-box access to) the prover, then the witness w can be extracted from it. Notice that soundness follows

from this demand, since if a witness can be extracted from the prover, then this witness apparently exists, so that $(x, w) \in R_L$ and thus $x \in L$.

The definition is as follows.

Definition 5.17 (Black-box zero-knowledge proof of knowledge). Let L be a language and let R_L be a polynomially computable relation for L. An interactive protocol between two interactive probabilistic polynomial-time algorithms $\mathcal{P}$ and $\mathcal{V}$ is a *black-box zero-knowledge proof of knowledge* for R_L ([GMR89], see also [CDM00; Gol00]) if there exists an expected polynomial-time simulator $\mathcal{S}$ and a polynomial-time extractor χ, that satisfy the following conditions:

Completeness For all x, w such that $R_L(x, w) = 1$,

$$\Pr[\mathcal{P}(x, w) \leftrightarrow \mathcal{V}(x) \to 1] = 1.$$

Black-box zero-knowledge For any probabilistic Turing machine $\mathcal{V}^*$ that is polynomial-time in its first argument, and for any auxiliary input $a \in \{0,1\}^*$, we have

$$\begin{aligned} &\{\text{view}_{\mathcal{V}^*}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x, a))\}_{x \in L,\, a \in \{0,1\}^*} \\ &\quad \overset{c}{\approx} \{\text{view}_{\mathcal{V}^*}(\mathcal{S}(x) \overset{\blacksquare}{\to} \mathcal{V}^*(x, a))\}_{x \in L,\, a \in \{0,1\}^*} \end{aligned} \tag{5.1}$$

for any w such that $(x, w) \in R_L$.

Black-box extraction Let $x \in L$. For any probabilistic Turing machine $\mathcal{P}^*$ that is polynomial-time in its first argument, and for any auxiliary input $a \in \{0,1\}^*$, if

$$\Pr[\mathcal{P}^*(x, a) \leftrightarrow \mathcal{V}(x) \to 1] \geq \epsilon(|x|)$$

for some function $\epsilon \colon \mathbb{N} \to [0, 1]$, then there exists a negligible function ν such that

$$\Pr[\mathcal{P}^*(x, a) \overset{\blacksquare}{\leftarrow} \chi(x) \to w : R(x, w) = 1] \geq \epsilon(|x|) - \nu(|x|).$$

In other words, if the prover convinces the verifier often that it knows some witness for $x \in L$, then the extractor computes a witness for x almost as often.

Notice the symmetry in the definition: even if the verifier cheats then it can learn nothing interesting, while even if the prover cheats then it cannot convince verifiers if it does not know a witness (and if neither cheats, then the protocol works as expected).

We will sometimes use that the second property, black-box zero-knowledge, implies (and is in fact equivalent to) the following: there exists a negligible function ν such that for any $x \in L$, any w such that $(x, w) \in R$, and any auxiliary input a:

$$\left| \Pr[\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x, a) \to 1] - \Pr[\mathcal{S}(x) \overset{\blacksquare}{\to} \mathcal{V}^*(x, a) \to 1] \right| = \nu(|x|). \tag{5.2}$$

That is, if using a witness w the prover can get verifier $\mathcal{V}^*$ to accept, then the simulator $\mathcal{S}$ can, given black-box access to the verifier, make him accept without knowing the witness w. Indeed, suppose some machine $\mathcal{V}^*$ violates the formula above, i.e., there exist x, w, a

such that with non-negligible probability $\mathcal{V}^*(x, a)$ outputs 0 when it interacts with $\mathcal{S}(x)$ and 1 when interacting with $P(x, w)$. Note that the output of $\mathcal{V}^*$ can be calculated in polynomial time from its view. Consider then the distinguisher that, given such a view, returns the output of $\mathcal{V}^*$ corresponding to this view. This distinguisher would then, for these particular x, w, a, be able to distinguish $\text{view}_{\mathcal{V}^*}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x, a))$ from $\text{view}_{\mathcal{V}^*}(\mathcal{S}(x)^{\blacksquare}\!\!\rightarrow \mathcal{V}^*(x, a))$, violating equation (5.1).

5.5.4 Σ-protocols and other variations

The definition above is stated in terms of black-box simulators: for *all* verifiers there exists *one* algorithm $\mathcal{S}$ that, when given black-box access to the possibly malicious verifier $\mathcal{V}^*$, interacts with $\mathcal{V}^*$ in a way that is indistinguishable from honest verifiers. Since such an oracle Turing machine with black-box access to $\mathcal{V}^*$ can be thought of as a Turing machine that depends on $\mathcal{V}^*$, this implies a more general phrasing: for *every* verifier $\mathcal{V}^*$ there exists a simulator $\mathcal{S}_{\mathcal{V}^*}$ whose interaction with $\mathcal{V}^*$ is indistinguishable from those of honest provers. This version is indeed more general than the black-box version, in the sense that not all zero-knowledge protocols are black-box [Bar01]. Such non-black-box proofs will however not occur in this thesis.

For most purposes it is sufficient that the honest and simulated views are indistinguishable by polynomial-time algorithms. When the two are not only indistinguishable but actually identically distributed, then the protocol is said to be *perfect* zero-knowledge.

The definition in the previous subsection guarantees security against dishonest verifiers by demanding that for any verifier (i.e., even one that deviates from the protocol), the simulator's behavior is indistinguishable from honest users. There is an important weaker variation in which the verifier is assumed to be *honest-but-curious* instead; that is, it follows the protocol but may retain and study all data that it received afterwards. Protocols that are secure against such verifiers are called honest-verifier zero-knowledge. Although they offer less security they are often much simpler, and therefore more efficient and easier to study. Additionally, there exist ways in which one can turn any Σ-protocol (see Definition 5.19 below) into a full zero-knowledge protocol [Dam10].

Definition 5.18 (Honest-verifier zero-knowledge). An interactive proof system for a language L and relation R_L is an interactive protocol between a prover $\mathcal{P}$ and verifier $\mathcal{V}$, such that

Completeness For any $(x, w) \in R_L$ and any $a \in \{0,1\}^*$, we have

$$\Pr[\mathcal{P}(x, w) \leftrightarrow \mathcal{V}(x, a) \to 1] = 1.$$

Soundness For any $x \notin L$, any interactive algorithm $\mathcal{P}^*$, and any $a, w \in \{0,1\}^*$, we have

$$\Pr[\mathcal{P}^*(x, w) \leftrightarrow \mathcal{V}(x, a) \to 1] \leq \frac{1}{3}.$$

(Note that this probability can be lessened simply by executing the protocol a number of times.)

Common information: $P_1, \ldots, P_n, K \in G$; the group G		
Prover		**Verifier**
knows k_i such that $K = \prod_{i=1}^{n} P_i^{k_i}$		
random $w_1, \ldots, w_n \in_R \mathbb{Z}_p^*$		
send $W = P_1^{w_1} \cdots P_n^{w_n}$	$\longrightarrow$	
	$\longleftarrow$	send $c \in_R \mathbb{Z}_p$
$\forall i \in [1, n]$ send $z_i = ck_i + w_i$	$\longrightarrow$	
		verify $K^c W \stackrel{?}{=} \prod_{i=1}^{n} P_i^{z_i}$

Figure 5.2. The Schnorr Σ-protocol for DL-representations.

An interactive proof system between a prover $\mathcal{P}$ and verifier $\mathcal{V}$ is *honest-verifier zero-knowledge* if there exists a probabilistic polynomial-time algorithm $\mathcal{S}$ such that for any w, the probability ensembles $\{\text{view}_{\mathcal{V}}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}(x))\}_{x \in L}$ and $\{\mathcal{S}(x)\}_{x \in L}$ are computationally indistinguishable.

Notice the absence of a quantifier over all possible verifiers $\mathcal{V}^*$: the output of the simulator $\mathcal{S}$ is required to be indistinguishable only from the views of honest verifiers. Clearly, black-box zero-knowledge proofs of knowledge are also honest-verifier zero-knowledge but not vice versa.

An important class of honest-verifier zero-knowledge protocols is the following.

Definition 5.19 (Σ-protocols). An honest-verifier zero-knowledge protocol is called a Σ-*protocol* if the following holds.

- It consists of three moves: first the prover sends a *commitment* W, then the challenger responds with a *challenge* c, after which the prover sends a reply z.
- For any x and any pair of accepting conversations (W, c, z), (W, c', z') for x with $c \neq c'$, one can efficiently compute a w such that $(x, w) \in R_L$ (this property, which can be seen as a variant of black-box extractability, is called *special soundness*).

5.5.5 Examples

Example 5.20 (Σ-protocol for DL-representations). Let G be a cyclic group of prime order p in which the DL-problem is intractable, and suppose the prover wants to prove knowledge of the DL-representation $(k_1, \ldots, k_n)$ of $K \in G$ with respect to the distinct non-unit elements $P_1, \ldots, P_n \in G$. Then the prover can follow the Σ-protocol in Figure 5.2 [Bra00]. If one takes $n = 1$ then it reduces to the Schnorr Σ-protocol for discrete logarithms [Sch90]. It is easy to see that this is indeed a Σ-protocol:

- Given $K \in G$ and base points $P_1, \ldots, P_n \in G$, one can generate an accepting conversation by taking arbitrary numbers $c, z_1, \ldots, z_n \in \mathbb{Z}_p$ and setting $W = K^{-c} \prod_{i=1}^{n} P_i^{z_i}$. Such conversations are distributed identically to actual traces for $K = \prod_{i=1}^{n} P_i^{k_i}$.

- If $(W, c, z_1, \ldots, z_n)$, $(W, c', z'_1, \ldots, z'_n)$ are two accepting traces of this protocol for K, then we apparently have

$$K^{-c} \prod_{i=1}^{n} P_i^{z_i} = W = K^{-c'} \prod_{i=1}^{n} P_i^{z'_i},$$

which when solved to K yields

$$K = \prod_{i=1}^{n} P_i^{(z'_i - z_i)/(c' - c)}.$$

This says that $k_i = (z'_i - z_i)/(c' - c)$ for $i = 1, \ldots, n$ is a valid witness for K.

Example 5.21 (BBZKPoK for DL-representations). Let G again be a cyclic group of prime order p in which the DL-problem is intractable. We present a black-box zero-knowledge proof of knowledge for proving knowledge of DL-representations in G that is an optimization of the one from [CDM00]. When $n = 1$ is taken, it reduces to a zero-knowledge proof of knowledge for discrete logarithms (although for that case [CDM00] contains a more efficient protocol).

Recall that a trace (W, c, z) of the Schnorr Σ-protocol for K is valid if $P^z = WK^c$. Such a trace can be generated by choosing z, c randomly, setting $W = P^z K^{-c}$, and returning (W, c, z). Suppose now that the prover $\mathcal{P}$ wants to prove knowledge of the DL-representation $(k_1, \ldots, k_n)$ of $K = P_1^{k_1} \cdots P_n^{k_n}$ with respect to $P_1, \ldots, P_n \in G$, where all P_i are distinct and unequal to 1. The proof consists of two parts, each having three moves:

- $\mathcal{V}$ uses the simulator above for K with respect to P_1, obtaining (W, c, z). It sends W to $\mathcal{P}$, and executes the Schnorr Σ-protocol on the numbers c, z such that $W = K^{-c} P_1^z$.
- $\mathcal{P}$ uses the OR-protocol from [CDS94] (a Σ-protocol consisting of 3 moves) to show that it knows either numbers c', z' such that (W, c', z') is accepting for K (with P_1 as base point), or the DL-representation $(k_1, \ldots, k_n)$ of K, without revealing which.

The second and third moves of the first part can then be executed simultaneously with the first and second moves of the second part, resulting in a protocol of 4 moves with $n + 6$ exponentiations for the prover. The protocol is shown in Figure 5.3; for readability the protocol has not been contracted there to 4 moves. We emphasize that in implementations, moves 2 and 4 and moves 3 and 5 should be done simultaneously.

We briefly sketch the existence of a simulator and extractor for this protocol. First we remark that the proofs from the first part and second part are both witness-hiding [FS90].

The simulator uses its black-box access to the verifier to extract the numbers c, z from the first part. Then it can use these in the OR-protocol in the second part to prove that it either knows (c, z) or $(k_1, \ldots, k_n)$. Since the verifier cannot detect that it is being rewinded and since the OR-protocol is witness-hiding, the verifier cannot tell which of the two statements the simulator is proving. Therefore the simulator's behavior is indistinguishable from honest provers.

Common information: $P_1, \ldots, P_n, K \in G$; the group G

Prover		**Verifier**
knows k_i such that $K = \prod_{i=1}^{n} P_i^{k_i}$		
	Phase 1	
$c_1, e, z_1', z_2', \bar{w}_1, \ldots, \bar{w}_n \in_R \mathbb{Z}_p^*$		$c, z, w_1, w_2 \in_R \mathbb{Z}_p^*$
	$\longleftarrow$	send $W = P_1^z K^{-c}, \tilde{W} = P_1^{w_1} K^{w_2}$
send e	$\longrightarrow$	
	$\longleftarrow$	send $z_1 = w_1 + ez, z_2 = w_2 - ec$
verify $P_1^{z_1} K^{z_2} \stackrel{?}{=} \tilde{W} W^e$		
	Phase 2	
send $W' = P_1^{z_1'} K^{z_2'} W^{-c_1}, \bar{W} = P_1^{\bar{w}_1} \cdots P_n^{\bar{w}_n}$	$\longrightarrow$	
	$\longleftarrow$	send c
set $c_2 = c - c_1$		
$\forall i$ set $\bar{z}_i = \bar{w}_i + c_2 k_i$		
send $c_1, c_2, z_1', z_2', (\bar{z}_i)_{i=1,\ldots,n}$	$\longrightarrow$	
		verify $c \stackrel{?}{=} c_1 + c_2$
		verify $P_1^{z_1'} K^{z_2'} \stackrel{?}{=} W' W^e$
		verify $P_1^{\bar{z}_1} \cdots P_n^{\bar{z}_n} \stackrel{?}{=} \bar{W} K^{c_2}$

Figure 5.3. Black-box zero-knowledge proof of knowledge of a DL-representation.

The extractor runs the first part of the protocol normally. Then, using its black-box access to $\mathcal{P}$ it can rewind $\mathcal{P}$ in the second part, and obtain either the DL-representation $(k_1, \ldots, k_n)$ of K (in which case we are done), or numbers c', z' such that (W, c', z') is accepting for K (with P_1 as base point). Suppose the latter is the case. Since the Σ-protocol executed in the first part is witness-hiding, the new trace (W, c', z') will with overwhelming probability differ from the the trace (W, c, z) that the extractor generated itself. Thus it sets $k' = (z - z')/(c - c')$, so that $K = P_1^{k'}$. Now the extractor outputs $(k', 0, \ldots, 0)$ (which is also a DL-representation of K with respect to $P_1, \ldots, P_n$).

Other black-box zero-knowledge proofs of knowledge that can be used for DL-representations are, for example, the one sketched in [Dam10, p. 16] which also has 4 moves, or the protocol for discrete logs from [Lin11, p. 31], which is easily extended to a protocol for DL-representations but has 5 moves.

5.5.6 Conventions and notations

We will use the Camenisch-Stadler notation [CS97] for zero-knowledge proofs. For example, if K, P_1, P_2 are elements of some group then

$$\mathrm{PK}\{(k_1, k_2) : K = P_1^{k_1} P_2^{k_2}\} \tag{5.3}$$

denotes a zero-knowledge proof of knowledge of the numbers k_1, k_2 that satisfy the relation $K = P_1^{k_1} P_2^{k_2}$. We will, however, not switch to Greek letters for the unknowns, as is sometimes done, but we will consistently write all unknowns (e.g. k_1 and k_2 in (5.3)) on the right-hand side.

5.6 Signature schemes

A *signature scheme* is a set of protocols dealing with authenticity of digital messages. A valid digital signature over some message should give a recipient (or *verifier*) reason to believe that the message was created (or at least seen and approved) by a known authority, who we shall call the *signer*. The purpose of a signature scheme is thus to use the trust relationship between the signer and a verifier to certify messages. Signature schemes are widely used throughout computer science and the internet, and they often also serve as basis for credential schemes. We will in fact create a signature scheme, and prove its unforgeability, for precisely that purpose in Section 9.2.3 on p. 157.

Signature schemes work as follows. First the signer creates two related pieces of information called the *secret key* and *public key*. It keeps the secret key to itself but publishes the public key. Then, given a message, the signer can create a digital signature over that message using the secret key. After this anyone who knows the signer's public key and who is given the message and signature is able to verify that that signature is valid over that particular message, using the public key.

The validity of a signature over a message with respect to a certain public key should ensure the verifier that the signature was created by the signer. This means that the trust between the verifier and signer can only be extended to the message if *only* the signer, who holds the secret key that corresponds to its known public key, is able to create signatures that will verify when using the public key. Therefore, when given only a public key and a message, but not the secret key, it should be infeasible to produce a signature over that message that will verify with the given public key (that is, the scheme should be *unforgeable*).

Let us now turn to the formal definition of signature schemes.

Definition 5.22 (Signature schemes). A *signature scheme* consists of the following three algorithms:

KeyGen(1^ℓ) Given a security parameter ℓ in unary, this algorithm outputs two bit strings PK, SK, as well as a description of a set $\mathcal{M}$ called the *message space*.

Sign$_{SK}(m)$ Given a secret SK key and message $m \in \mathcal{M}$, this algorithm outputs a signature σ.

Verify$_{PK}(\sigma, m)$ Given a public key PK, signature σ and message m, this algorithm outputs `valid` or `invalid`.

The scheme should be *correct*, in the following sense: for every $(SK, PK, \mathcal{M}) \leftarrow \mathsf{KeyGen}(1^\ell)$ and every message $m \in \mathcal{M}$, we have

$$\mathsf{Verify}_{PK}(\mathsf{Sign}_{SK}(m), m) = \texttt{valid}.$$

Notice that this definition says nothing about unforgeability. For example, the scheme where $\mathsf{Sign}_{SK}(m) = 0$ and $\mathsf{Verify}_{PK}(\sigma, m) = \texttt{valid}$ for all SK, PK, m, σ satisfies the definition but is clearly not very useful. The definition above only specifies the *syntax* of signature schemes. As is often done in cryptography, we separately define the security notion in terms of a *security game*: a process consisting of a number of fixed moves, in which we endow an interactive algorithm with some capabilities that it may use as it sees fit. We call this algorithm the *adversary*. Its goal is to use these capabilities to output a valid signature that was not created by the signer. If there exists no adversary that can do this, then our signature scheme is unforgeable. There are many versions of this game, differing in what capabilities they give to the adversary, and when the adversary is considered to have won the game. The one that is most often used is due to Goldwasser, Micali and Rackoff [GMR88], and goes as follows.

Definition 5.23. The existential unforgeability game under adaptive chosen message attacks for a signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is a game between an adversarial user $\mathcal{A}$ and a signer $\mathcal{S}$, controlled by the challenger, and proceeds as follows.

Setup The challenger generates a private-public key pair $(SK, PK) = \mathsf{KeyGen}(1^\ell)$. It sends PK and the description of the message space to the adversary $\mathcal{A}$.

Queries The adversary requests signatures on messages $m_1, \ldots, m_q \in \mathcal{M}$ that it may choose adaptively (i.e., message m_i may depend on messages $m_1, \ldots, m_{i-1}$). The challenger responds to each query with a signature $\sigma_i \leftarrow \mathsf{Sign}_{SK}(m_i)$.

Output The adversary $\mathcal{A}$ outputs a pair (m, σ) and wins the game if σ is a valid signature over m, and $m \neq m_i$ for all $1 \leq i \leq q$.

When no probabilistic polynomial-time algorithm can win this game with non-negligible probability in ℓ, then we say that the signature scheme is existentially unforgeable under adaptive chosen-message attacks.

The signature scheme is said to be *nondeterministic* when the Sign algorithm uses randomness, so that two signings of the same message may result in different signatures. Notice that if the Sign algorithm is deterministic, then the secret key SK and message m completely determine the resulting signature σ. If the Sign algorithm is probabilistic, however, then it may return different signatures if it is run twice on the same secret key and message. In such cases, the win condition for the adversary is sometimes modified as follows: the adversary wins if $(m, \sigma) \neq (m_i, \sigma_i)$ for all i. That is, it wins if it manages to output *any* new signature, even if it is over an already-seen message. This is sometimes called *strong* unforgeability. Although this stronger notion of unforgeability can sometimes be desirable, the weaker game leaves open the possibility to modify valid signatures over some message into a new valid signature over the same message. In

particular, self-blindable credential schemes (see Section 5.7 and Chapter 6) critically depend on this ability.

Sometimes a similar game is used in which the adversary must present all of the messages that it wants signed at once, instead of being allowed to choose them adaptively. If no adversary can win this game then the scheme is said to be weakly unforgeable under (non-adaptively chosen) chosen-message attacks. It is clear that this is a weaker notion of unforgeability, in the sense that any scheme which is unforgeable under adaptively-chosen message attacks is also unforgeable in this sense, but not necessarily vice versa. Nevertheless, this weaker notion will play an important role in Chapter 7 (see Definition 7.3 on p. 128).

Example 5.24 (BLS signatures). Consider the following deterministic signature scheme, by Boneh, Lynn and Shacham [BLS04] but modified to Type 3 pairings [SV07b].

KeyGen(1^ℓ) Generate a bilinear group pair $e\colon G_1 \times G_2 \to G_T$ of Type 3 (see Definition 5.11) where the order p of the three groups has length ℓ, together with a *hash function*[13] $H\colon \{0,1\}^* \to G_1$. Additionally, generate a number $a \in \mathbb{Z}_p^*$ and a generator $Q \in G_2$, and set $A = Q^a$. The public key is the description of e, G_1, G_2, G_T and H, along with $A, Q \in G_2$. The private key is a. The message space is $\mathcal{M} = \{0,1\}^*$.

Sign$_a(m)$ Given a message $m \in \{0,1\}^*$, the signature is $\sigma = H(m)^a \in G_1$.

Verify$_A(\sigma, m)$ The signature σ over the message m is valid only if

$$e(\sigma, Q) = e(H(m), A).$$

The correctness of this scheme is clear. Indeed, if $\sigma = H(m)^a$ then

$$e(\sigma, Q) = e(H(m)^a, Q) = e(H(m), Q^a) = e(H(m), A).$$

One can prove that the scheme is unforgeable under adaptively chosen message attacks under a generalization of the the CDH assumption (Definition 5.9) to Type 3 pairings, in the random oracle model.[14]

5.7 Credential schemes

We now turn to the main topic of this part of the thesis: *credential schemes*. We have seen in the previous section that signature schemes serve to use the trust relationship between

[13] A function $H\colon \{0,1\}^* \to G_1$ is a hash function if it is efficiently computable, and if it satisfies the following properties. Given a $h \in G_1$ it should be infeasible to find a bitstring x such that $h = H(x)$ (preimage resistance); given a bitstring x_1 it should be infeasible to find a bitstring x_2 such that $H(x_1) = H(x_2)$ (second preimage resistance); and it should be infeasible to find two bitstrings x_1 and x_2 such that $H(x_1) = H(x_2)$ (collision resistance). Since none of our own constructions in this thesis rely on hash functions, we will not spend more attention on them here.

[14] A *random oracle model* is an idealized hash function that when given a message m for the first time returns a random element from G, and then consistently returns the same element when queried on the same message m. In the unforgeability proof, the adversary is given oracle access to this oracle, and it is controlled by the challenger. Since our schemes will not use hash functions, we will have no need of this model (in which case we say that we operate in the *standard model*).

a signer and a verifier in order to certify a message. In credential schemes there is also such a trusted party (that we will now call the *issuer*), but the goal of credential schemes is to extend the trust relationship between the issuer and verifier not to a message, but to a *user* (for example, a person, or a computing device acting on behalf of a person). Thus, a user can obtain a credential from an issuer, after which he can show it to a verifier. The scheme should be such that if the verifier finds that the user's credential is valid, then the verifier can be sure that the user indeed obtained this credential from the issuer. Since the issuer apparently deemed this user worthy of a credential, the verifier can trust the user just as far as it trusts the issuer.

One thing that is immediately clear is that whatever process we use to show a credential to a verifier is going to have to be more sophisticated than just handing over the credential to the verifier (as is usually done in the case of a signature), because this would allow the verifier to present the credential it just obtained to other verifiers (i.e., the verifier could *impersonate* the user). If this would be possible then the second verifier could no longer be sure that it is talking to an honest user instead of to an impersonator, so that no trust relationship can be established. This suggests that in order to prevent these replay attacks, the user and verifier should engage in some interactive protocol that hides the credential at least partially, so that the verifier cannot replay the credential to others.

Notice also that like in signature schemes there is the following asymmetry: only the issuer should be able to create credentials, while any verifier should be able to verify their validity (as long as the user agrees to try to convince the verifier of this fact). Therefore we will here too need private and public keys.

Apart from its validity, a credential may or may not contain information about its owner. We might call schemes whose credentials do not contain such information *boolean credential schemes* (since their credentials are either valid or not). We will study a number of such schemes in Chapter 6. By contrast, *attribute-based credentials* (ABC's) can contain not one but several pieces of information (called *attributes*). The showing protocol of these schemes is then such that the user can choose to show some of these attributes, while hiding the others. This ability makes them very suitable for the kind of flexibility and privacy-friendliness that we are looking for in identity-management schemes (see p. xii). These schemes will be the topic of the final three chapters of this thesis.

Throughout this thesis we will write n for the maximum number of attributes that an (instantiation of an) attribute-based scheme allows. Note that a boolean credential scheme can be seen as an attribute-based credential schemes that allows zero attributes, i.e., $n = 0$. For this reason we will in the remainder of this section focus on attribute-based credential schemes; by taking $n = 0$ one obtains the definition and security notions for boolean schemes.[15] (The exception to this will be the unforgeability game; we will use a separate notion of unforgeability for boolean schemes called *malleability*, see Definition 6.4 on p. 116. For readability we will in Chapter 6, which is exclusively concerned with boolean schemes, repeat the relevant definitions and security games without reference to n or to attributes.)

Summarizing the discussion so far, we can now define credential schemes as follows.

[15]Notice, however, that even though the definitions below contain an n it will not make sense to try to use a nonzero n for boolean schemes, and in the case of attribute-based schemes the security proofs may collapse if one takes $n = 0$.

Definition 5.25 (Attribute-based credential schemes). An attribute-based credential scheme consists of the following protocols. (We assume a single issuer, but this can easily be generalized to multiple issuers.)

KeyGen $(1^\ell, n)$ This algorithm takes as input a security parameter ℓ and the number of attributes n that the credentials will contain, and outputs the issuer's private key SK and public key PK, which must contain the number n, and a description of the attribute space M.

Issue An interactive protocol between an issuer $\mathcal{I}$ and user $\mathcal{P}$ that results in a credential c:

$$\mathcal{I}(PK, SK, (k_1, \ldots, k_n)) \leftrightarrow \mathcal{P}(PK, k_0, (k_1, \ldots, k_n)) \to c.$$

Here k_0 is the user's private key, that is to be chosen from the attribute space M by the user; the Issue protocol should prevent the issuer from learning it. We assume that before execution of this protocol, the issuer and user have reached agreement on the values of the attributes $k_1, \ldots, k_n$. The secret key and attributes $k_0, k_1, \ldots, k_n$ are contained in the credential c.[16]

ShowCredential An interactive protocol between a user $\mathcal{P}$ and verifier $\mathcal{V}$ which is such that, if c is a credential issued using the Issue protocol over attributes $(k_1, \ldots, k_n)$ using private signing key SK corresponding to public key PK, then for any disclosure set $\mathcal{D} \subset \{1, \ldots, n\}$ the user can make the verifier accept:

$$\mathcal{P}(PK, c, \mathcal{D}) \leftrightarrow \mathcal{V}(PK, \mathcal{D}, (k_i)_{i \in \mathcal{D}}) \to 1.$$

We assume that here, too, the user has notified the verifier in advance of the disclosure set $\mathcal{D}$ and disclosed attributes $(k_i)_{i \in \mathcal{D}}$.

Notice that this syntax is fairly close to that of signature schemes: in both cases a private-public key pair is generated; and in both cases the private one is used for the generation of the object at hand, while the public one is used for verification. Because of this similarity we can define unforgeability of credential schemes in a way that closely resembles that of signature schemes (see the next section). Additionally, most of the credential schemes in the literature consist of an underlying signature scheme, along with an interactive protocol for issuing them and another for showing them. In fact, the schemes that we will define in Chapters 7 and 9 also have this structure.

Some credential schemes additionally allow credentials to be *revoked* if they are lost of stolen. In this case, there is also a revocation authority, and an interactive protocol called Revoke between the user and the revocation authority. Additionally, during the ShowCredential protocol the user must also convince the verifier that the credential that he is presenting has not been revoked using the Revoke protocol. We will, however, not concern ourselves with revocation in this thesis; instead we refer to [Lue+15] for a revocation scheme that is suitable for attribute-based credential schemes such as those from Chapters 7 and 9.

[16] We realize that in terms of terminology, it is rather unusual to include the secret key in the credential. We do this mainly for notational convenience in the later chapters.

Apart from the properties already mentioned, we expect any credential scheme to satisfy the following properties.

- *Unforgeability* (see the next section): no user can prove ownership of a credential or of attributes that have not been issued to it by the issuer.
- *Offline issuer*: The issuer is not involved in the verification of credentials.
- *Selective disclosure* for attribute-based schemes: any subset of attributes contained in a credential can be disclosed.

Some schemes additionally are such that multiple uses of the same credential cannot be linked; this is called *unlinkability*. We will discuss this in more detail in Section 5.7.3 below.

Example 5.26. Consider the following simple example of a boolean scheme (that does not offer unlinkability).

KeyGen(1^ℓ) The issuer generates private and public keys for some signature scheme that is unforgeable under adaptively chosen message attacks. Additionally, it generates a cyclic group G of order p with $|p| = \ell$, in which the DL-problem is intractable, along with a generator $P \in G$ (by using the BLS signature scheme from Example 5.24 on p. 103 the group G can also serve as the set of signatures).

Issue The user generates some $k \in_R \mathbb{Z}_p$ and sends $K = P^k$ to the issuer, and performs

$$\mathrm{PK}\{(k) \colon K = P^k\}$$

using a zero-knowledge proof of knowledge (see Section 5.5). If the proof is successful, the issuer signs K and sends the resulting signature to the user.

ShowCredential The user sends K and the signature to the verifier, and performs

$$\mathrm{PK}\{(k) \colon K = P^k\}$$

to prove knowledge of his private key k. The verifier accepts if the proof is successful and if the signature over K is valid.

Because the user hides his secret key k from the verifier using the zero-knowledge proof, the verifier is prevented from replaying the credential to others.

5.7.1 Conventions and notations

As mentioned above, we will write n for the amount of attributes that a scheme allows, and we will index the attributes with the letter i. When discussing multiple credentials (particularly in the unforgeability and unlinkability games and proofs) we index credentials with j, and write m for the amount of credentials that are currently under consideration. We will use the letter k for attributes. Thus, $k_{i,j}$ would denote the i-th attribute of the j-th credential.

In the case of attribute-based credentials, we will write $\mathcal{D} \subset \{1, \ldots, n\}$ for the index set of the disclosed attributes, and

$$\mathcal{C} = \{1, \ldots, n\} \setminus \mathcal{D}$$

for the index set of the undisclosed attributes. The index 0 of the secret key k_0 is not considered part of this set, as k_0 is always kept secret.

5.7.2 Unforgeability

We define unforgeability of an attribute-based credential scheme in terms of the following game (notice the resemblance with the game from Definition 5.23).

Definition 5.27 (Unforgeability game for ABC's). The unforgeability game of an attribute-based credential scheme between a challenger and an adversary $\mathcal{A}$ is defined as follows.

Setup For a given security parameter ℓ, the adversary decides on the number of attributes $n \geq 1$ that each credential will have, and sends n to the challenger. The challenger then runs the $\mathsf{KeyGen}(1^\ell, n)$ algorithm of the credential scheme and sends the resulting public key to the adversary.

Queries The adversary $\mathcal{A}$ can make the following queries to the challenger.

- **Issue**$(k_{1,j}, \ldots, k_{n,j})$ The challenger and adversary engage in the Issue protocol, with the adversary acting as the prover and the challenger acting as the issuer, over the attributes $(k_{1,j}, \ldots, k_{n,j})$. It may choose these adaptively.
- **ShowCredential**$(\mathcal{D}, k_1, \ldots, k_n)$ The challenger creates a credential with the specified attributes $k_1, \ldots, k_n$, and engages in the ShowCredential protocol with the adversary, acting as the prover and taking $\mathcal{D}$ as disclosure set, while the adversary acts as the verifier.

Challenge The challenger, now acting as the verifier, and the adversary, acting as the user, engage in the ShowCredential protocol. The adversary chooses a disclosure set $\mathcal{D}$, and if it manages to make the verifier accept then it wins if one of the following holds:

- If the adversary made no Issue queries then it wins regardless of the disclosure set (even if $\mathcal{D} = \emptyset$);
- Otherwise $\mathcal{D}$ must be nonempty, and if $(k_i)_{i \in \mathcal{D}}$ are the disclosed attributes, then there must be no j such that $k_i = k_{i,j}$ for all $i \in \mathcal{D}$ (i.e., there is no single credential issued in an Issue query containing all of the disclosed attributes $(k_i)_{i \in \mathcal{D}}$).

We say that the credential scheme is *unforgeable* if no probabilistic polynomial-time algorithm can win this game with non-negligible probability in the security parameter ℓ.

5.7.3 Unlinkability

One feature that a credential scheme may or may not offer, and that will be of particular interest to us, is that of unlinkability. A scheme is said to offer *multi-show unlinkability* if whenever two credentials are shown (disclosing the same attributes with the same values, in the case of attribute-based schemes), then the verifier cannot tell whether he

was shown one and the same credential twice, or two distinct credentials. In the case of boolean schemes, this means that the user is completely anonymous within the set of all other users of the scheme; in the case of attribute-based schemes the anonymity set is the set of users who have credentials containing the same attributes as the disclosed ones. For example, if a user discloses an attribute saying "I am over 18 years old", then he will be anonymous within the set of all people that also have this attribute.

Another possible kind of unlinkability that a credential scheme may offer (separately or simultaneously with multi-show unlinkability) is *issuer unlinkability*. Consider a user whose credential contains certain attributes. If this user then engages in the ShowCredential protocol with the issuer, or with a verifier who sends the trace of the run to the issuer afterwards, then of all credentials that the issuer issued with those attributes, the issuer cannot tell which one was used. That is, the user is from the perspective of the issuer anonymous within the set of people whose credentials have the same values for the disclosed attributes.

We now define a single unlinkability game. Afterwards, we will argue that this game implies both kinds of unforgeability.

Definition 5.28 (Unlinkability game for ABC's). The unlinkability game of an attribute-based credential scheme between a challenger and an adversary $\mathcal{A}$ is defined as follows.

Setup For a given security parameter ℓ, the adversary decides on the number of attributes $n \geq 1$ that each credential will have, and sends n to the challenger. The adversary then runs the $\mathsf{KeyGen}(1^\ell, n)$ algorithm from the credential scheme and sends the resulting public key to the challenger.

Queries The adversary $\mathcal{A}$ can make the following queries to the challenger.

- **Issue**$(k_{1,j}, \ldots, k_{n,j})$ The adversary chooses a set of attributes $(k_{1,j}, \ldots, k_{n,j})$, and sends these to the challenger. Then, acting as the issuer, the adversary engages in the Issue protocol with the challenger, issuing a credential j to the challenger having attributes $(k_{1,j}, \ldots, k_{n,j})$.
- **ShowCredential**$(j, \mathcal{D})$ The adversary and challenger engage in the showing protocol on credential j, the challenger acting as the user and the adversary as the verifier. Each time the adversary may choose the disclosure set $\mathcal{D}$.
- **Corrupt**(j) The challenger sends the entire internal state, including the secret key k_0, of credential j to the adversary.

Challenge The adversary chooses two uncorrupted credentials j_0, j_1 and a (possibly empty) disclosure set $\mathcal{D} \subset \{1, \ldots, n\}$. These have to be such that the disclosed attributes from credential j_0 coincide with the ones from credential j_1, i.e., $k_{i,j_0} = k_{i,j_1}$ for each $i \in \mathcal{D}$. It sends the indices j_0, j_1 and $\mathcal{D}$ to the challenger, who checks that this holds; if it does not then the adversary loses.
Next, the challenger flips a bit $b \in_R \{0, 1\}$, and acting as the user, it engages in the ShowCredential with the adversary on credential j_b. All attributes whose index is in $\mathcal{D}$ are disclosed.

Output The adversary outputs a bit b' and wins if $b = b'$.

We define the advantage of the adversary $\mathcal{A}$ as $\mathsf{Adv}_\mathcal{A} := |\Pr[b = b'] - 1/2|$. When no

probabilistic polynomial-time algorithm can win this game with non-negligible advantage in the security parameter ℓ, then we say that the credential scheme is *unlinkable*.

In this game the adversary plays the role of the issuer in the Setup phase and the role of the verifier in the Challenge phase. This definition of unlinkability implies both multi-show unlinkability and issuer unlinkability, as follows.

Multi-show unlinkability Suppose there exists a malicious verifier that can link two transactions as in the Challenge phase of our unlinkability game, without itself having issued the credentials from those transactions. Then there certainly also exists an adversary that, by using this verifier, breaks blindness in the sense of Definition 5.28. Thus unlinkability as in Definition 5.28 implies multi-show unlinkability.

Issuer unlinkability Consider the following game for issuer unlinkability. The Setup and Queries phases are as in Definition 5.28, but the Challenge and Output phases are as follows:

Challenge The adversary chooses a credential j and a disclosure set $\mathcal{D} \subset \{1, \ldots, n\}$, and informs the challenger of its choice. The challenger flips a bit $b \in_R \{0,1\}$, takes $j_0 \in_R \{1, \ldots, m\}$, and sets $j_1 = j$. Next it engages in the ShowCredential protocol with the adversary on credential j_b, acting as the user. All attributes whose index is in $\mathcal{D}$ are disclosed.

Output The adversary outputs a bit b' and wins if $b = b'$.

If there exists an adversary that can win this game, then there also exists an algorithm that breaks blindness in the sense of Definition 5.28: if an algorithm can win this game with non-negligible probability, it means that it can distinguish credential $j = j_1$ from any other credential $j \in_R \{1, \ldots, m\}$, so that it could certainly also distinguish j_1 from a fixed j_0.

Essentially, the reason why these variations of the unlinkability game imply unlinkability in the sense of Definition 5.28, is because in both of them the adversary is endowed with less power. Indeed, in the first case it does not know the issuer's secret key, and since it did not issue the credentials it does not have access to the issuer's view of the executions of the Issue protocol; and in the second case it does not get to choose the credential j_1.

Chapter 6

Linkability and malleability in self-blindable credentials

Self-blindable credential schemes allow users to anonymously prove ownership of credentials. This is achieved by randomizing the credential before each showing in such a way that it still remains valid. As a result, each time a different version of the same credential is presented. A number of such schemes have been proposed, but unfortunately many of them are broken, in the sense that they are *linkable* (i.e., failing to protect the privacy of the user), or *malleable* (i.e., they allow users to create new credentials using one or more valid credentials given to them). In this chapter we prove a general theorem that relates linkability and malleability in self-blindable credential schemes, and that can test whether a scheme is linkable or malleable. After that we apply the theorem to a number of self-blindable credential schemes to show that they suffer from one or both of these issues.

This chapter is based on the following article.

[HLR15] J.-H. Hoepman, W. Lueks, and S. Ringers. "On Linkability and Malleability in Self-blindable Credentials". In: *Information Security Theory and Practice: 9th IFIP WG 11.2 International Conference, WISTP 2015*. Ed. by N. R. Akram and S. Jajodia. Cham: Springer International Publishing, 2015, pp. 203–218.

The main theorem of this chapter and the corresponding article is my own work, based on an initial, rough idea by Jaap-Henk Hoepman. My own contribution was to extend and formalize this idea in the form of the theorem; to apply it to the credential schemes that we treat in this chapter; and to write the text itself, with help from both Wouter Lueks and Jaap-Henk Hoepman.

6.1 Introduction

Unlinkable credential schemes (sometimes also called anonymous credential schemes) are a promising technique for secure and privacy-friendly identity management. They are given by an issuer to the user, who can then prove possession of it to other parties. This showing should be multi-show unlinkable: that is, it is infeasible for the issuer, the verifier or any other party to determine whether two transactions did or did not originate from the same user. Additionally, credentials have to be unforgeable, in the sense that the user cannot create his own credential, or modify one or more existing ones in order to obtain a new credential (this kind of forgeability is called *malleability* and plays and important role in this chapter). A number of such systems already exist; we mention, for example, Idemix [CL01; IBM12]. This scheme is additionally attribute-based, meaning that a credential may contain multiple attributes (which are pieces of information or statements, generally about the owner of the credential). Such systems tend to be complex, however, which is why considerable effort has gone into simpler credential systems that have no attributes (for example [HJV10]; see also Example 6.2). Instead, such (boolean) credentials are either valid or invalid, resulting in simpler constructions that are easier to study and potentially more efficient, allowing for practical implementations of such credentials on smart cards. Naturally, such credential schemes still have to be unlinkable and unforgeable.

A simple boolean credential scheme is that of Example 5.26, where the user knows (so can prove knowledge of) the private key corresponding to a public key that has been signed by the issuer. A problem with this scheme, however, is that the user presents the same public key and signature on each use, making all uses of the same credential linkable. One technique for preventing such linkability is to modify the credential before each showing, in such a way that it remains valid. This is called *blinding*, and credential schemes that use this technique are called *self-blindable credential schemes*. The first example of such a scheme was given by Verheul in the same paper that defines the notion of self-blindability [Ver01]. The advantage of blinding credentials in such a way is that it is easy for the user (blinding is usually cheap) and for the verifier (verifying a blinded signature is generally not much different from verifying an ordinary signature).

In the past decade, a number of such self-blindable credential schemes have been proposed [CL01; EMO09; HJV10; KT08; Ver01]. Unfortunately, many of them are broken, in the sense that transactions are linkable or the credentials are malleable, or even both. In this chapter we uncover a common theme in the cause of the problem of each of these schemes: the dependence of the public key and signature on the private key of the credential can often be exploited to achieve linkability or malleability. This suggests there is a trade-off between the two. After having introduced and defined the relevant concepts in Section 6.2, we show this by proving a general theorem in Section 6.3 that makes it easy to determine whether a self-blindable credential scheme is linkable. The theorem exhibits an interesting and strong relationship between linkability and malleability of the credential scheme. We then apply this theorem in Section 6.4 to show that several proposed self-blindable schemes in the literature are linkable, and present explicit counter-examples as well. The theorem also indicates in which directions to look for self-blindable credential schemes that are both unlinkable and unmalleable.

6.2 Self-blindable credentials

In all self-blindable credential schemes that we know of, a credential consists of a private key k, a corresponding public key K, and a signature S over the public key that the issuer gives to the owner of the credential. That is, a credential C is of the form

$$C = (k, K, S) \in \mathcal{P} \times \mathcal{K} \times \mathcal{S}$$

where $\mathcal{P}$, $\mathcal{K}$ and $\mathcal{S}$ are the sets of private keys, public keys and signatures, respectively. We shall write $\mathcal{C}$ for the product $\mathcal{C} = \mathcal{P} \times \mathcal{K} \times \mathcal{S}$. Let us say that an element $(k, K, S) \in \mathcal{P} \times \mathcal{K} \times \mathcal{S}$ is *valid* when k is the private key corresponding to K and S is a valid signature over K with respect to the issuer's signing key.

Self-blindable credentials, introduced by Verheul [Ver01], are credentials that the user modifies each time before he shows it to a verifier, in such a way that it remains valid, and such that multiple transactions cannot be linked to each other. We define this notion as follows.[1]

Definition 6.1. A credential scheme is called *self-blindable* if

1. There exists a blinding-factor space $\mathcal{B}$ and an efficiently computable map

   $$B\colon \mathcal{C} \times \mathcal{B} \to \mathcal{K} \times \mathcal{S},$$

   such that if the credential $C = (k, K, S) \in \mathcal{C}$ is valid, then for any $\alpha \in \mathcal{B}$ it holds that if $B(C, \alpha) = (\bar{K}, \bar{S})$ then $\bar{S}$ is a valid signature over $\bar{K}$;
2. In the ShowCredential protocol, the credential C is blinded to $(\bar{K}, \bar{S}) = B(C, \alpha)$ for a random $\alpha \in_R \mathcal{B}$, after which $\bar{K}$ and $\bar{S}$ are used as the public key and signature respectively in the remainder of the ShowCredential protocol.

Most self-blindable credential schemes that we know of have a ShowCredential protocol of the following form:

1. The user blinds K and S using the blinding map B and sends the blinded values $\bar{K}$, $\bar{S}$ to the verifier, who then non-interactively checks that $\bar{S}$ is a valid signature over $\bar{K}$.
2. Afterwards, the user and verifier engage in a (possibly zero-knowledge) proof in which the user convinces the verifier that he knows the private key k and blinding factor α from which he calculated $\bar{K}$ (i.e., the first element from the tuple $(\bar{K}, \bar{S}) = B((k, K, C), \alpha)$).

We purposefully do not include the private key in the blinded credentials (that is, we do not demand that $B(C, \alpha) = (\bar{k}, \bar{K}, \bar{S})$, where $\bar{k}$ is the private key corresponding to $\bar{K}$), because if such a map B were to exist then anyone can, given one credential, create arbitrary new ones. That is, there would be no distinction between the creation of new

[1] In [Ver01], Verheul puts four extra demands on the blinding map B besides item 1 in our definition, that are meant to exclude edge cases that could never lead to desirable properties in a credential schemes. Instead of including these four extra properties, we describe the role of the blinding map B more directly in the second item in Definition 6.1.

credentials by the issuer and blinding an existing credential. In terms of Definition 6.4, the system would then be 1-malleable.

Example 6.2. As a first example we consider the self-blindable credential by Hoepman et al. [HJV10], which is based on the original scheme by Verheul [Ver01]. Here we use the Chaum–Pedersen [CP93] signature scheme, as follows. Consider an (additively written) Type 1 pairing $e\colon G_1 \times G_2 \to G_T$, with all groups of prime order p, and take generators P and Q for G_1 and G_2 respectively. Then the private signing key of the issuer is a number $a \in \mathbb{Z}_p$, and the corresponding public key is $A = aQ \in G_2$.

The space of private keys of credentials is $\mathcal{P} = \mathbb{Z}_p$, and for a private key $k \in \mathbb{Z}_p$ the corresponding public key is $K = kP \in G_1$. The signature on K is then a Chaum–Pedersen signature $S = aK$, which can be verified by

$$e(K, A) \stackrel{?}{=} e(S, Q).$$

Thus, we have $\mathcal{K} = \mathcal{S} = G_1$.

Blinding the public key is done by multiplying it by a random number $\alpha \in \mathbb{Z}_p$, that is, $\bar{K} = \alpha kP$, and similarly for the signature: $\bar{S} = \alpha akP$. The verification equation then becomes

$$e(\bar{K}, A) \stackrel{?}{=} e(\bar{S}, Q).$$

If $\bar{K}$ and $\bar{S}$ are blinded by the same value $\alpha \in \mathbb{Z}_p$, and if the unblinded signature is a valid Chaum–Pedersen signature over the unblinded public key K, then this equation holds.

The problem of this system is not linkability, but malleability. Given a credential (k, K, S) on its private key k the user can easily create a new credential $(\alpha k, \alpha K, \alpha S)$ on any other private key αk. This means that a user that has access to the internals of his credential can create a new credential over any private key $\bar{k} \in \mathbb{Z}_p$, without involving the issuer. (Hoepman et al. mitigate this attack by storing the private key on a smart card, so that the user cannot access it directly. It is, however, still a problem, for example because revocation in such a system would be impossible, because there is nothing that binds the private key to the user.)

We will examine this form of forgeability more closely in Definition 6.4 and Example 6.10. In this case, it is a consequence of the linearity of the Chaum–Pedersen signature S in the private key k. Later on, in Example 6.8, we will see how using a signature scheme that is nonlinear in k results in linkability.

This chapter is mostly concerned with how the blinded public key $\bar{K}$ and blinded signature $\bar{S}$ depend on the private key k and blinding factor α. Taking the blinded public key $\bar{K}$, we will denote the dependency of $\bar{K}$ on k by writing

$$\bar{K} = \mathsf{PubKey}(k, \alpha)$$

for a certain function $\mathsf{PubKey}\colon \mathcal{P} \times \mathcal{B} \to \mathcal{K}$. Similarly,

$$\bar{S} = \mathsf{Sig}_{SK}(k, \alpha).$$

for a certain function $\mathsf{Sig}_{SK}\colon \mathcal{P} \times \mathcal{B} \to \mathcal{S}$. Here SK is the issuer's private key. Using these functions PubKey and Sig_{SK}, we can express the blinding map B as follows:

$$B((k, K, S), \alpha) = (\bar{K}, \bar{S}) = (\mathsf{PubKey}(k, \alpha), \mathsf{Sig}_{SK}(k, \alpha)) .$$

We stress that these functions PubKey and Sig_{SK} need not correspond to any algorithm that is run by one of the involved parties (typically, for example, the user will calculate the blinded public key using the unblinded public key, not directly from the private key). The purpose of these functions is purely to make the dependence on the private key and blinding factor explicit.

6.2.1 Security properties

Having defined the basic structures and the notion of self-blindability, we next turn to the security properties that we expect boolean credential schemes to satisfy. If we adapt the unlinkability game for attribute-based schemes of Definition 5.28 to boolean schemes by letting the amount of attributes be 0, we obtain the following game.

Definition 6.3 (Unlinkability). A boolean self-blindable credential scheme is *unlinkable* if no adversary can win the following game with non-negligible advantage.

Setup The adversary runs the $\mathsf{KeyGen}(1^\ell)$ algorithm and sends the public key to the challenger.

Queries For any $j \in \{1, \ldots, m\}$ the adversary may issue the following queries:

- **Issue** The adversary, acting as the issuer, engages in the Issue protocol with the challenger, issuing a credential to the challenger that we label with j.
- **ShowCredentials**(j) The adversary acts as the verifier in the ShowCredential protocol for the credential j which has been issued in an Issue query, with the challenger acting as the user. The adversary sees the same interaction as a normal verifier would see.
- **Corrupt**(j) The adversary requests the credential j from an Issue query to be corrupted. The challenger gives him the internal state (k_j, K_j, S_j) of credential j.

Challenge The adversary selects two uncorrupted credentials j_0, j_1 from the set $\{1, \ldots, m\}$ and informs the challenger of his choice. The challenger then picks a bit $b \in_R \{0, 1\}$ at random, and runs ShowCredential on credential j_b with the adversary playing the role of the user while the adversary acts as the verifier. The adversary outputs a bit b'. He wins if $b = b'$.

This definition of linkability includes a weaker notion of linkability where the adversary only gets to see two traces, and has to decide whether they belong to the same user. Given such an algorithm $\mathcal{A}'$ we can then build an adversary $\mathcal{A}$ satisfying the definition above by having it perform the following actions:

Setup $\mathcal{A}$ sets up the unlinkability game with his challenger.

Queries $\mathcal{A}$ performs m Issue queries. Next he chooses two credentials j_0 and j_1 at random from the list of credentials $\{1, \ldots, m\}$ and performs a ShowCredential query on j_0. He stores the trace of the protocol run.

Challenge $\mathcal{A}$ informs his challenger that he has chosen the credentials j_0 and j_1 from the previous phase. He engages in the ShowCredential protocol on j_b (where b is chosen by the challenger) and stores the trace. Then, he uses the algorithm $\mathcal{A}'$ to compare the traces from j_0 and j_b. If $\mathcal{A}'$ returns that j_0 and j_b have the same public key then $\mathcal{A}$ outputs $b' = 0$ as his guess; otherwise he outputs $b' = 1$.

Then the adversary $\mathcal{A}$ satisfies Definition 6.3.

Definition 6.4 (m-malleability). Let $\{(k_1, K_1, S_1), \ldots, (k_m, K_m, S_m)\} \in \mathcal{C}^m$ be a tuple of m valid credentials. If there exists an efficiently computable map $F\colon \mathcal{C}^m \to \mathcal{C}$ which outputs a valid credential on a new private key (that is, if

$$(k, K, S) = F\Big((k_1, K_1, S_1), \ldots, (k_m, K_m, S_m)\Big)$$

and (k, K, S) is valid and $k \neq k_j$ for all $j = 1, \ldots, m$) then we say that the credential scheme is *m-malleable*.

Although malleability is nothing more than a particular kind of forgeability, it warrants a separate definition because it occurs in a number of existing credential schemes, and because it plays an important role in the theorem below. The problem that the definition above aims to capture is that new credentials can be made without the involvement or knowledge of the issuer, if the user has m credentials. We see that the credential scheme from Example 6.2 has 1-malleability: in that scheme, given a credential (k, K, S) and any $\alpha \in \mathbb{Z}_p$, the credential $(\alpha k, \alpha K, \alpha S)$ is a new valid credential. This is a problem, because the blinded credential should still be bound to the original private key k.

Note, however, that if the scheme is not attribute-based but boolean, then malleability is not necessarily a problem. Modifying an existing credential into a new one does not change any of its key properties: it was valid and it remains valid, so nothing has really changed. On the other hand, we can think of the following cases in which it would be a problem.

- The public key K may contain meaningful information such as attributes (as is the case in, for example, U-Prove). In this case, the user should not be able to manipulate this meaningful data, so it should be impossible by exploiting the malleability to obtain a new valid credential whose public key contains different information. In particular, the user should not be able to create a credential whose public key is $\bar{K}$ when given a credential with public key K.
- In a self-blindable credential scheme that is not attribute-based (for example the one from Example 6.2), issuers may issue multiple credentials (signed by different keys) instead of a single credential with multiple attributes. For example, a public key signed with private key a_1 may mean that the user is over 18, while one signed with private key a_2 could mean that he is a German citizen. In such a setting it should be impossible to combine credentials issued to different users. In this case, an underage German citizen should not be able to use his foreign friend's over 18

credential to prove that he is both over 18 and a German citizen. Normally, such a proof would show that the signed public keys in both credentials are identical, thus preventing credentials from being combined. However, malleability might make it possible to change a credential over one public key (say the foreign friend's) into another public key (say of the underage German citizen). This would make credential pooling trivial.
- Similarly, the unchecked randomization of the signed public key can make revocation, often an essential feature of anonymous credential systems, next to impossible.

In the next section, we show that malleability has a strong link with linkability, and then examine a number of credential schemes that suffer from these issues.

6.3 Relating malleability and linkability

In the credential schemes considered in this chapter, the public key $K \in \mathcal{K}$ depends linearly on the private key $k \in \mathcal{P}$. Any signature over K obviously depends on K, and therefore also on k. Thus, when considering suitable signature schemes, if the set of signatures is a group then we may take one that is either linear or not linear in k. The theorem and its corollary below then say the following: if the signature scheme is *not* linear in k, then there is linkability, while if it is linear in k then the scheme *may* be malleable. Loosely speaking, this is because if the public key and the signature do not depend on the user's private key and the blinding factor in precisely the same way, then this can be exploited. Let us now make this more precise.

We assume henceforth that $\mathcal{K}$ and $\mathcal{S}$ are groups, that we will write additively, and that $\mathcal{P}$ is a field. From the corollary below and onwards it will, moreover, be the case that the former two are vector spaces[2] over $\mathcal{P}$, meaning that elements from $\mathcal{K}$ and $\mathcal{S}$ can be multiplied on the left by elements from $\mathcal{P}$: for example, $kK \in \mathcal{K}$ for $k \in \mathcal{P}$ and $K \in \mathcal{K}$. We recall the following definition.

Definition 6.5. A map $L\colon V \to W$, with V and W being vector spaces over $\mathcal{P}$, is *linear* if $L(v+v') = L(v) + L(v')$ and $L(kv) = kL(v)$ for all $v, v' \in V$ and $k \in \mathcal{P}$.

We denote with $\mathsf{Verify}_{PK}\colon \mathcal{K} \times \mathcal{S} \to \{\mathsf{true}, \mathsf{false}\}$ the verification function of the signature scheme under consideration, where PK is the public key of the issuer. That is, Verify_{PK} is such that

$$\mathsf{Verify}_{PK}\left(\mathsf{PubKey}(k,\alpha), \mathsf{Sig}_{SK}(k,\alpha)\right) = \mathsf{true}$$

for all k, α. On the other hand, whenever $k \neq k'$ or $\alpha \neq \alpha'$ (or both), we should have

$$\mathsf{Verify}_{PK}\left(\mathsf{PubKey}(k,\alpha), \mathsf{Sig}_{SK}(k',\alpha')\right) = \mathsf{false}.$$

[2]The theorem remains true when $\mathcal{P}$ is not a field but a ring, in which case $\mathcal{K}$ and $\mathcal{S}$ are not vector spaces, but modules over $\mathcal{P}$. As this is the case in none of the examples below, however, we will keep calling $\mathcal{P}$ a field and $\mathcal{K}$ and $\mathcal{S}$ vector spaces for definiteness in the remainder of the chapter.

Theorem 6.6. *Consider a self-blindable credential scheme. Suppose that for each $k, k' \in \mathcal{P}$ and $\alpha, \alpha' \in \mathcal{B}$ there exist $\ell \in \mathcal{P}$ and $\beta \in \mathcal{B}$ such that*

$$\mathsf{PubKey}(k, \alpha) + \mathsf{PubKey}(k', \alpha') = \mathsf{PubKey}(\ell, \beta). \tag{6.1}$$

If Sig_{SK} also has this property for the same ℓ, β, that is,

$$\mathsf{Sig}_{SK}(k, \alpha) + \mathsf{Sig}_{SK}(k', \alpha') = \mathsf{Sig}_{SK}(\ell, \beta) \tag{6.2}$$

but only *when $k = k'$, then there is linkability. On the other hand, if Sig_{SK} has this property for any k, k', and*

- *the* ShowCredential *protocol allows the user to present $(\ell, \mathsf{PubKey}(\ell, \beta), \mathsf{Sig}_{SK}(\ell, \beta))$ as a valid credential,*
- *the user can efficiently compute ℓ and β,*

then there is 2-malleability.

Proof. Assume that Sig_{SK} has the stated property only when $k = k'$, and that equation (6.1) holds for any k, k'. Then if $k = k'$ we have $\mathsf{Sig}_{SK}(k, \alpha) + \mathsf{Sig}_{SK}(k', \alpha') = \mathsf{Sig}_{SK}(\ell, \beta)$ and similarly for PubKey, so

$$\begin{aligned}&\mathsf{Verify}_{PK}\left(\mathsf{PubKey}(k, \alpha) + \mathsf{PubKey}(k', \alpha'),\ \mathsf{Sig}_{SK}(k, \alpha) + \mathsf{Sig}_{SK}(k', \alpha')\right)\\ &\quad = \mathsf{Verify}_{PK}(\mathsf{PubKey}(\ell, \beta),\ \mathsf{Sig}_{SK}(\ell, \beta)) = \mathsf{true}.\end{aligned}$$

On the other hand, if $k \neq k'$, then $\mathsf{Sig}_{SK}(k, \alpha) + \mathsf{Sig}_{SK}(k', \alpha')$ does not evaluate to $\mathsf{Sig}_{SK}(\ell, \beta)$. Therefore

$$\begin{aligned}&\mathsf{Verify}_{PK}\left(\mathsf{PubKey}(k, \alpha) + \mathsf{PubKey}(k', \alpha'),\ \mathsf{Sig}_{SK}(k, \alpha) + \mathsf{Sig}_{SK}(k', \alpha')\right)\\ &\quad = \mathsf{false}.\end{aligned}$$

Thus, the function Verify_{PK} returns true when applied to the sum of the two credentials involved if and only if $k = k'$, so that the scheme is linkable.

The second part of the statement is obvious: if the ShowCredential protocol does not prevent the user from using $(\ell, \mathsf{PubKey}(\ell, \beta), \mathsf{Sig}_{SK}(\ell, \beta))$ as a valid credential then he can present it to verifiers, even though $\mathsf{Sig}_{SK}(\ell, \beta)$ was not given to him by the issuer. □

Corollary 6.7. *Suppose the function* PubKey *is linear in both arguments. If Sig_{SK} is linear in the second but not the first argument, then there is linkability. If Sig_{SK} is linear in both arguments, then there is 1-malleability.*

Proof. Suppose Sig_{SK} is linear in the second but not the first argument, and that $k = k' \in \mathcal{P}$. Then

$$\begin{aligned}&\mathsf{PubKey}(k, \alpha) + \mathsf{PubKey}(k', \alpha')\\ &\quad = \mathsf{PubKey}(k, \alpha) + \mathsf{PubKey}(k, \alpha')\\ &\quad = \mathsf{PubKey}(k, \alpha + \alpha'),\end{aligned}$$

and since Sig_{SK} is also linear in the second argument, we will also have $\mathsf{Sig}_{SK}(k,\alpha) + \mathsf{Sig}_{SK}(k',\alpha') = \mathsf{Sig}_{SK}(k,\alpha+\alpha')$. Thus $\mathsf{Sig}_{SK}(k,\alpha+\alpha')$ will be a valid signature over $\mathsf{PubKey}(k,\alpha+\alpha')$.

On the other hand, if $k \neq k'$ then

$$\begin{aligned}
&\mathsf{PubKey}(k,\alpha) + \mathsf{PubKey}(k',\alpha') \\
&\qquad = k\mathsf{PubKey}(1,\alpha) + k'\mathsf{PubKey}(1,\alpha') \\
&\qquad = k\alpha\mathsf{PubKey}(1,1) + k'\alpha'\mathsf{PubKey}(1,1) \\
&\qquad = (k\alpha + k'\alpha')\mathsf{PubKey}(1,1) \\
&\qquad = \mathsf{PubKey}(k\alpha + k'\alpha', 1),
\end{aligned}$$

but now $\mathsf{Sig}_{SK}(k,\alpha) + \mathsf{Sig}_{SK}(k',\alpha') \neq \mathsf{Sig}_{SK}(k\alpha + k'\alpha', 1)$, because Sig_{SK} is not linear in its first argument. Hence the verification function Verify_{PK} over the sum of both credentials will distinguish $k = k'$ and $k \neq k'$, so that the credential scheme is linkable.

Concerning the second statement of the corollary, if both PubKey and Sig_{SK} are linear in both arguments, then

$$\mathsf{PubKey}(k,\alpha) = \alpha\mathsf{PubKey}(k,1) = \mathsf{PubKey}(\alpha k, 1),$$

and similarly $\mathsf{Sig}_{SK}(k,\alpha) = \mathsf{Sig}_{SK}(\alpha k, 1)$, so that $(\alpha k, \mathsf{PubKey}(k,\alpha), \mathsf{Sig}_{SK}(k,\alpha))$ is a valid credential. Therefore, there is 1-malleability. □

Essentially, the corollary implies that when the verification function is used directly in the ShowCredential protocol, then it is very difficult to assure that it is neither linkable nor malleable. Indeed, if the public key is linear in the private key while the signature is not, then there is likely linkability through the verification equation of the signature scheme. On the other hand, if they are both linear in the private key then it is likely that the system suffers from malleability.

In spite of this difficulty it is certainly possible to create self-blindable credential schemes that are neither malleable nor linkable; we will discuss this in more detail in Section 6.5. In the next section, we discuss a number of self-blindable credential schemes, that all suffer from one of these problems.

6.4 Broken self-blindable credential schemes

Example 6.8. For this example we reuse the PubKey function from Example 6.2, but this time we use the weak Boneh–Boyen signature scheme instead (see Section 7.2 on p. 130 and [BB08]). In this scheme the public and private keys of the issuer are $a \in \mathbb{Z}_p$ and $A = aQ \in G_2$ respectively, as before. A signature on $k \in \mathbb{Z}_p$ is $S = \frac{1}{a+k}P$. Setting $K = kQ \in G_2$ (note that now $K \in G_2$, contrary to Example 6.2), the signature S may be verified by checking that $e(S, A+K) \stackrel{?}{=} e(P,Q)$.

We still blind the public key and signature by multiplying it with a random number

User		Verifier
choose blinding $\alpha \in_R \mathbb{Z}_p$		
send $\alpha K, \alpha S, \alpha A, \alpha P, \alpha Q$	$\longrightarrow$	into $\bar{K}, \bar{S}, \bar{A}, \bar{P}, \bar{Q}$
		verify $e(\bar{S}, \bar{A} + \bar{K}) \stackrel{?}{=} e(\bar{P}, \bar{Q})$
$\mathrm{PK}\{(\kappa) : \bar{K} = \kappa P\}$	$\longleftrightarrow$	

Figure 6.1. Self-blindable credential scheme from Example 6.2 modified to use the Boneh–Boyen signature scheme.

α, i.e.,

$$\bar{K} = \mathsf{PubKey}(k, \alpha) = \alpha k Q$$

and

$$\bar{S} = \mathsf{Sig}_a(k, \alpha) = \frac{\alpha}{a + k} P.$$

In addition, the user will also have to send $\bar{A} = \alpha A$, $\bar{P} = \alpha P$ and $\bar{Q} = \alpha Q$ to the verifier. The verification is done by checking

$$e(\bar{S}, \bar{A} + \bar{K}) \stackrel{?}{=} e(\bar{P}, \bar{Q}).$$

The ShowCredential protocol of this scheme might look as in Figure 6.1.

In this case, if $k = k'$ then $\mathsf{PubKey}(k, \alpha) + \mathsf{PubKey}(k', \alpha') = \mathsf{PubKey}(k, \alpha + \alpha')$, and similarly $\mathsf{Sig}_a(k, \alpha) + \mathsf{Sig}_a(k', \alpha') = \mathsf{Sig}_a(k, \alpha + \alpha')$, so that Verify_A will return true. On the other hand, if $k \neq k'$, then

$$\mathsf{PubKey}(k, \alpha) + \mathsf{PubKey}(k', \alpha') = \mathsf{PubKey}(\alpha k + \alpha' k', 1)$$

while

$$\mathsf{Sig}_a(k, \alpha) + \mathsf{Sig}_a(k', \alpha') \neq \mathsf{Sig}_a(\alpha k + \alpha' k', 1)$$

so Verify_A will return false. Thus, this system is linkable.

Example 6.9. Like the scheme from Example 6.2, the self-blindable credential scheme from Kiyomoto and Tanaka [KT08] uses Chaum-Pedersen signatures, but this time on a Type 1 curve (i.e., $G_1 = G_2 = G$ and $P = Q$). The issuer's public key is $A = aP$.

The private key here consists of two numbers $(\kappa, \kappa') \in \mathbb{Z}_p^2$, where κ is random while $\kappa' = m\kappa$ is a *non-repudiation private key*; here m is a number encoding some valuable piece of information related to the user. This would discourage users from sharing their credential, because if another party learns κ and κ' then it could recover m. Setting $k := \kappa + \kappa'$, the corresponding public key and signature are $K = kP$ and $S = aK = akP$. The ShowCredential protocol of this scheme is shown in Figure 6.2.

This scheme suffers from a number of problems. First, the relation $k = \kappa + m\kappa$ is

User		Verifier
choose blinding $\alpha \in_R \mathbb{Z}_p$		
send $\alpha K, \alpha S$	$\longrightarrow$	into $\bar{K}, \bar{S}$
		verify $e(\bar{K}, A) \stackrel{?}{=} e(\bar{S}, P)$
		choose nonce $\eta \in_R \mathbb{Z}_p$
into N	$\longleftarrow$	send ηP
send $\alpha\kappa N, \alpha\kappa' N$	$\longrightarrow$	into $\bar{M}, \bar{M}'$
		verify $e(\bar{M} + \bar{M}', P) \stackrel{?}{=} e(\bar{K}, \eta P)$
		run $\mathsf{RevocationCheck}(\bar{K}, \bar{M})$

Figure 6.2. Self-blindable credential scheme by Kiyomoto et al. [KT08] (simplified).

User		Verifier
		choose nonce $\eta \in_R \mathbb{Z}_p$
into N	$\longleftarrow$	send ηQ
choose blinding $\alpha \in_R \mathbb{Z}_p$		
send $\alpha S, \alpha k N, \alpha A, \alpha N, \alpha P$	$\longrightarrow$	into $\bar{S}, \bar{K}, \bar{A}, \bar{Q}, \bar{P}$
		verify $e(\bar{P}, A) = e(P, \bar{A})$
		verify $e(\bar{P}, \eta Q) = e(P, \bar{Q})$
		verify $e(\bar{S}, \eta\bar{A} + \bar{K}) = e(\bar{P}, \bar{Q})$
		run $\mathsf{RevocationCheck}(\bar{A}, \bar{K}, \bar{Q})$

Figure 6.3. Self-blindable credential scheme by Emura et al. [EMO09] (simplified).

nowhere enforced by the ShowCredential protocol, in the sense that the user could use $\lambda, k - \lambda$ for some random $\lambda \in \mathbb{Z}_p$ instead of κ, κ'. This means that users can easily share credentials after all, without fear of disclosing the valuable information encoded by m.

Second, without going into the details of the revocation mechanism, we remark that it relies on how k splits into $k = \kappa + \kappa'$, so that the problem above allows users to present revoked credentials without problems. (In addition, the revocation mechanism introduces linkability.)

Third, since both the public key K and signature S are linear in both the blinding factor α and private key k, by Corollary 6.7 the scheme is 1-malleable. For any α and valid credential $((\kappa, \kappa'), K, S)$ the user can present the credential $((\lambda, \alpha k - \lambda), \alpha K, \alpha S)$. (Actually, because the public key $A = aP \in G$ lives in the same group as the signatures $S = akP \in G$, anyone can easily create his own credential by setting $K = (\kappa + \kappa')P =: kP$ for some random $\kappa, \kappa' \in \mathbb{Z}_p$, and $S = kA$ – that is, the system is actually 0-malleable.)

Example 6.10. Some of the problems of the credential scheme above were pointed out by Emura et al. [EMO09], who came up with an improved protocol that we will examine in this example. In this protocol the malleability is solved through the use of Boneh–Boyen signatures. Theorem 6.6 shows, however, that it is linkable. We explain the problem here.

The ShowCredential protocol is shown in Figure 6.3. As in Example 6.8, the Boneh–

Boyen signature is of the form

$$\left(A, P, Q, S = \frac{1}{a+k}P\right),$$

where $A = aQ$ and $K = kQ$. We include the values A, P and Q explicitly in the signature because these are blinded as well in the ShowCredential protocol. The blinding factor is (η, α), where η is chosen by the verifier and α by the user. The blinded signature is then $(\alpha A, \alpha P, \alpha\eta Q, \alpha S)$, while the blinded public key is $\alpha\eta K$.

Theorem 6.6 is directly applicable to this scheme; we now describe the resulting linkability attack. Suppose the ShowCredential protocol is executed twice, and let (η_j, α_j) be the blinding factors used in two runs of the ShowCredential protocol, for $j = 1, 2$. Let $\bar{A}_j, \bar{P}_j, \bar{Q}_j, \bar{S}_j, \bar{K}_j$ be the values that the user sends to the issuer. We take the sum of two traces as follows:

$$\begin{aligned}
\bar{A} &= \eta_1\bar{A}_1 + \eta_2\bar{A}_2 = (\alpha_1\eta_1 + \alpha_2\eta_2)A, \\
\bar{P} &= \bar{P}_1 + \bar{P}_2 = (\alpha_1 + \alpha_2)P, \\
\bar{Q} &= \bar{Q}_1 + \bar{Q}_2 = (\alpha_1\eta_1 + \alpha_2\eta_2)Q, \\
\bar{S} &= \bar{S}_1 + \bar{S}_2 = \alpha_1 S_1 + \alpha_2 S_2, \\
\bar{K} &= \bar{K}_1 + \bar{K}_2 = \alpha_1\eta_1 K_1 + \alpha_2\eta_2 K_2.
\end{aligned} \tag{6.3}$$

Now we put these values in the third verification equation as follows:

$$e(\bar{S}, \bar{A} + \bar{K}) \stackrel{?}{=} e(\bar{P}, \bar{Q}). \tag{6.4}$$

If $K_1 = K_2 =: K$, then also $S_1 = S_2 =: S$ holds, and the lower two equations of (6.3) become $\bar{S} = (\alpha_1 + \alpha_2)S$, $\bar{K} = (\alpha_1\eta_1 + \alpha_2\eta_2)K$. Then equation (6.4) will hold. On the other hand, if $K_1 \neq K_2$ then equation (6.4) will not hold. Thus, transactions are linkable by the third verification equation.[3]

6.5 Do unmalleable, unlinkable self-blindable credential schemes exist?

Let us briefly consider a number of ways in which the pitfall outlined by Theorem 6.6 might be avoided. Suppose first that both the public key and the signature are linear in

[3]Note, however, that only the verifier can calculate the element $\bar{A} = \eta_1\bar{A}_1 + \eta_2\bar{A}_2$ (which is needed in order to perform this linking attack), as it contains η_1, η_2 which are never sent to the user. This differs from the linkability described in Example 6.8, in which anyone that can eavesdrop on the communication between the user and verifier can execute the attack. On the other hand, transactions can also be linked by checking the following equation:

$$e(\alpha_1 P, \alpha_2 S_2) = e(\bar{P}_1, \bar{S}_2) \stackrel{?}{=} e(\bar{P}_2, \bar{S}_1) = e(\alpha_2 P, \alpha_1 S_1)$$

which will hold if and only if $S_1 = S_2$; that is, when the signatures are the same. This attack can be done by any eavesdropper.

the private key, and that the sum of a trace of the ShowCredential protocol again constitutes a valid public key and signature. Then this can only be abused by a malicious user if he is able to calculate the corresponding private key. Therefore, if this is not feasible (perhaps because the private key k can only be calculated by the issuer, or because not all private keys are valid or allowable), then the system would not be malleable in the sense of Definition 6.4.

As another approach, one might take a public key and signature scheme that are both nonlinear in the private key k, or both nonlinear in the blinding factor α. In that case neither of the statements of Theorem 6.6 would be applicable. Going further, the ShowCredential protocol may be such that it is not necessary to send the public key to the verifier at all, so that it can play no role in either linkability or malleability. (This approach is taken in Idemix (see Example 6.11 below), and our own credential scheme in Chapter 9.)

Finally, it is important to note that we have so far only considered deterministic signature schemes (this is in fact implicitly enforced by our notations in this chapter). Using nondeterministic signature schemes intuitively makes more sense for the purpose of creating unlinkability, since in that case there no longer is a 1-to-1 correspondence between the public key and its signature that could be exploited to link credentials. This, too, is done both in Idemix and in our own scheme in Chapter 9.

Example 6.11. The Idemix credential scheme [CL01; IBM12] is an attribute-based credential scheme which is neither linkable nor malleable, and indeed, Proposition 6.6 does not apply to Idemix. This is because the ShowCredential protocol is substantially different from the ones of the other schemes discussed so far. In short, it goes as follows: the user partially blinds the Camenisch–Lysyanskaya [CL02] signature (A, e, v), resulting into $(\bar{A}, e, \bar{v})$, and sends $\bar{A}$ to the verifier. After that, they engage in an interactive zero knowledge-proof in which the user shows that he knows e, $\bar{v}$, and his private key, without disclosing any of these. This has the following consequences:

- There is no clear separation between the sending and verification of the public key and signature on the one hand, and a proof of knowledge of the secret key on the other hand. Both of these happen in a single interactive algorithm.
- In fact, the user does not directly send the public key to the verifier at all, blinded or otherwise. As a result, the map PubKey does not play any role in the ShowCredential algorithm.
- The map Sig_{SK} is not linear in the blinding factor.

The first two points will also apply to our credential scheme in Chapter 9. For more on Idemix, see also Remark 9.9 on p. 159.

Summarizing, self-blindable schemes that are both unmalleable and unlinkable certainly do exist: our scheme in Chapter 9 is an example – and if we take a slightly broader view of self-blindability then Idemix is an example as well. The margin for error seems to be small, however.

6.6 Conclusion

Creating a self-blindable credential scheme which is neither malleable nor linkable is hard, and indeed all self-blindable credential schemes that we have studied in this chapter are broken. There is a common theme in their failures: the use of the verification equation of the signature scheme in the ShowCredential protocol may cause linkability or malleability. We believe that this observation in the form of Theorem 6.6 and Corollary 6.7, together with the examples showing the consequences of this observation, will be of help in the creation of new, secure and anonymous self-blindable credential schemes.

Chapter 7

Partially blind Boneh–Boyen signatures

In a partially blind signature scheme, one part of the message as well as the resulting signature is hidden from the signer, while the other part of the message is visible to the signer. In this chapter we present a partially blind signature scheme that is closely related to Boneh–Boyen signatures, and prove its security in the standard model using only standard hardness assumptions. As an application, we introduce a single-show attribute-based credential scheme with short signatures.

An early version of the signing protocol of the signature scheme and the security proofs is due to Wouter Lueks. My contribution was to extend the protocol to support partial blindness, to formalize the security proofs, the single-show attribute-based credential scheme, and the majority of the text below, with many helpful improvements and suggestions from Wouter Lueks and Jaap-Henk Hoepman.

7.1 Introduction

A blind signature scheme allows a signer to sign a message without learning the message or the resulting signature. First introduced by Chaum [Cha83], blind signatures are used in many privacy-sensitive applications, like electronic cash (e.g., [CFN90]), electronic voting (e.g., [JC97]), and unlinkable credentials (e.g., [Bra00; Cha90]).

In these applications the blind signatures hide private information, for example the user's identity. The signer has no control over this information. However, in some scenarios the signer needs to add information to the signature that it does control, for example an expiry date. In these cases one could use a *partially* blind signature scheme. These schemes were introduced by Abe and Fujisaki [AF96] and further formalized by

Abe and Okamoto [AO00]. In such schemes one part of the message remains hidden but another part is visible to the signer. The known part of the message, called *common information*, can then be used for information that the signer and user agree upon in advance.

The contribution of this chapter is twofold. Our first contribution is a partially blind signing protocol for a signature scheme that is closely related to the Boneh–Boyen signature scheme [BB08], together with a full security proof in the standard model, using only standard hardness assumptions. If no common information is included, our scheme reduces to a blind signature scheme for actual Boneh–Boyen signatures. This is the first protocol for blindly signing Boneh–Boyen signatures (although a number of schemes exist for closely related signature schemes; we review some of these in Section 7.6). The Boneh–Boyen signature scheme produces very short signatures, and its security does not rely on the random oracle model (ROM). This scheme has many applications, for example, in attribute-based signatures [MPR11], group signatures [Gro07], and verifiable random functions [DY05].

Our second contribution is a (single-show) attribute-based credential scheme that makes use of our partially blind signature scheme. Our scheme is similar to U-Prove [Bra00; PZ13], but unlike U-Prove, which does not have an unforgeability proof [BL13b], our scheme is provably unforgeable in the standard model (i.e., without relying on the random oracle model).

We first describe the Boneh–Boyen signature scheme itself in Section 7.2, after which we define our partially blind signature scheme in Section 7.3. In Section 7.4, we prove that our scheme is indeed partially blind and unforgeable. Next, in Section 7.5 we present our single-show attribute-based credential scheme as a first example of our partially blind signature scheme. Finally, in Section 7.6 we compare our scheme with a number of other schemes from the literature.

In this chapter we use elements from $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$ for multiple values of n. For notational convenience we temporarily deviate from our previous conventions in this chapter, and identify $\mathbb{Z}_n$ with $\{0, \dots, n-1\}$, that is, $a \in \mathbb{Z}_n$ refers to the smallest nonnegative representative of the equivalence class a mod n (instead of to the equivalence class itself). If m is a different integer and $a \in \mathbb{Z}_n$, this will allow us to write a mod m without ambiguity. (Alternatively, we can consider "mod m" to be a map from $\mathbb{Z}_n$ to $\mathbb{Z}_m$ that always acts on the lowest representative of its argument, and that happens to be a ring homomorphism when m divides n.)

7.1.1 Partially blind signature schemes

A partially blind signature scheme [AO00; Cha83] is made up of three algorithms: KeyGen, Sign, and Verify, for generating keys, signing messages, and verifying signatures, respectively. These algorithms work as follows:

KeyGen(1^ℓ) Given a security parameter ℓ, this algorithm outputs a random key pair (SK, PK) together with a description of the message space $\mathcal{M}$ and the space of common information $\mathcal{I}$.

Sign This is an interactive protocol between the signer $\mathcal{S}$ and a user $\mathcal{U}$ that results in a

signature σ on the hidden message $m \in \mathcal{M}$ and common information $k \in \mathcal{I}$:

$$\mathcal{S}(SK, k) \leftrightarrow \mathcal{U}(PK, m, k) \rightarrow \sigma.$$

Verify$_{PK}(m, k, \sigma)$ This algorithm takes a public key PK, a message $m \in \mathcal{M}$, common information $k \in \mathcal{I}$, and a signature σ. It returns `valid` or `invalid`.

This differs from the usual syntax of signature schemes (see Definition 5.22 on p. 101) in that the Sign algorithm is interactive, and that the Verify_{PK} algorithm also takes the common information k as argument.

The signature scheme should be correct, in the sense that a signature σ over (m, k) signed with SK should verify with PK; i.e., if $(SK, PK) \leftarrow \mathsf{KeyGen}(1^\ell)$ and $\mathcal{S}(SK, k) \leftrightarrow \mathcal{U}(PK, m, k) \rightarrow \sigma$ then $\mathsf{Verify}_{PK}(m, k, \sigma)$ outputs `valid`.

In the next two games, we define the security notions for partially blind signature schemes. The blindness game follows the one by Abe and Okamoto [AO00], although we have slightly simplified it by allowing the adversary full control over the common information k.

Definition 7.1 (Blindness). We define blindness of a partially blind signature scheme in terms of the following game. It is a game between an adversarial signer $\mathcal{A}$ and the challenger who acts as two users. It proceeds as follows.

Setup Adversary $\mathcal{A}$ runs $(SK, PK) \leftarrow \mathsf{KeyGen}(1^\ell)$, also obtaining the descriptions of the spaces $\mathcal{M}$ and $\mathcal{I}$. It chooses common information $k \in I$ and two messages $m_0, m_1 \in \mathcal{M}$, and sends the description of $\mathcal{M}$ and $\mathcal{I}$ along with PK, m_0, m_1 and k to the challenger.

Run The challenger chooses a bit $d \in_R \{0, 1\}$, and simulates two users $\mathcal{U}_0$ and $\mathcal{U}_1$. It gives m_d and $m_{\bar{d}}$ (where $\bar{d} = 1 - d$) to $\mathcal{U}_0$ and $\mathcal{U}_1$ respectively, and lets them perform the signing protocol with the adversary.

Result If both users $\mathcal{U}_0$ and $\mathcal{U}_1$ output valid triples (m_d, k, σ_d) and $(m_{\bar{d}}, k, \sigma_{\bar{d}})$ respectively, then the challenger sends σ_0 and σ_1 to the adversary.

Guess Adversary $\mathcal{A}$ outputs his guess $d' \in \{0, 1\}$. It wins if $d = d'$ and the signatures σ_0 and σ_1 are valid.

We say that a signer ϵ-breaks the blindness of a signature scheme if it can win this game with advantage ϵ.

Notice that the game guarantees unlinkability only if the common information k is the same in both signing sessions, otherwise the adversary could use k to link the signature with its view of the Sign protocol.

Our unforgeability game follows that of Abe and Okamoto [AO00], with the exception that in our game the adversary has full control over the common information k. It is clear that if no such adversary exists, then an adversary that allows the challenger to partly control k also cannot exist.

Definition 7.2 (Unforgeability under chosen-message attacks). We define unforgeability under chosen-message attacks of a partially blind signature scheme in terms of the following game. It is a game between an adversarial user $\mathcal{A}$ and a signer $\mathcal{S}$, controlled by the challenger. The game proceeds as follows.

Setup The challenger generates a private-public key pair $(SK, PK) \leftarrow \mathsf{KeyGen}(1^\ell)$. It sends PK along with the descriptions of $\mathcal{M}$ and $\mathcal{I}$ to the adversary $\mathcal{A}$.

Queries The adversary and challenger engage in the Sign protocol over at most q message-pairs $(m_j, k_j) \in \mathcal{M} \times \mathcal{I}$ that are chosen adaptively by the adversary. In each query the adversary sends k_j to the challenger, while keeping m_j hidden.

Output For any $k \in \mathcal{I}$, denote with q_k the number of successfully completed queries in which the common information was k (if k occurred in none of the queries, set $q_k = 0$). The adversary $\mathcal{A}$ outputs some $k \in \mathcal{I}$, together with $q_k + 1$ pairs (m_j, σ_j), and wins the game if each σ_j is a valid signature over (m_j, k), and $(m_{j_1}, \sigma_{j_1}) \neq (m_{j_2}, \sigma_{j_2})$ for all $1 \leq j_1, j_2 \leq q_k + 1, j_1 \neq j_2$.

We say that our signature scheme is (t, q, ϵ)-existentially unforgeable under chosen message attacks if there exists no probabilistic polynomial-time algorithm that can win the above game with probability ϵ, running in time at most t and making at most q signature queries. In this game the adversary can let each message on which it queries the challenger depend on the public key, and on the previous messages. Notice that it suffices for the adversary to output just one extra pair (m, σ); for example, it could be that σ is a new signature over an already seen message m_j, or perhaps the message is new but $\sigma = \sigma_j$ for some j. In these cases, the adversary still wins.

7.1.2 Weakly unforgeable signature schemes

We will reduce the unforgeability (in terms of the game above) of our partially blind signature scheme to the unforgeability of weak Boneh–Boyen signatures. This signature scheme satisfies a weaker form of unforgeability than unforgeability under adaptively-chosen message attacks (see Definition 5.23 on p. 102), in the sense that the adversary has to send the messages that it wants signed before it receives the public key; in particular, it cannot choose them adaptively. The game is as follows [GMR88].

Definition 7.3 (Weak unforgeability under chosen message attacks). We define weak unforgeability under chosen message attacks of a (non-blind) signature scheme in terms of the following game. It is a game between an adversarial user $\mathcal{A}$ and a signer $\mathcal{S}$, which is controlled by the challenger. The game proceeds as follows.

Announcement The adversary $\mathcal{A}$ announces at most q messages $m_1, \ldots, m_q \in \mathcal{M}$ that it wants signed.

Response The challenger generates a private-public key pair $(SK, PK) \leftarrow \mathsf{KeyGen}(1^\ell)$. It generates q signatures $\sigma_i \leftarrow \mathsf{Sign}_{SK}(m_i)$ over the messages m_i that $\mathcal{A}$ chose earlier. It sends PK and the signatures σ_i to $\mathcal{A}$.

Output The adversary $\mathcal{A}$ outputs a pair (m, σ) and wins the game if σ is a valid signature over m, and $(m, \sigma) \neq (m_i, \sigma_i)$ for all $1 \leq i \leq q$.

We say that our signature scheme is (t, q, ϵ)-weakly existentially unforgeable under chosen message attacks if there exists no probabilistic polynomial-time algorithm that can win the above game with probability ϵ, running in time at most t and sending at most q messages in the announcement phase.

7.1.3 Paillier encryption

Our interactive signing protocol uses Paillier encryption, so we briefly introduce this scheme here. Recall that a public-key encryption scheme consists of three probabilistic polynomial-time algorithms $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ together with a message space $\mathcal{M}$, where

- $\mathsf{KeyGen}(1^\ell)$ Given a security parameter ℓ, this algorithm outputs a private key SK, a public key PK, and a description of the message space $\mathcal{M}$.
- For every $(SK, PK) \leftarrow \mathsf{KeyGen}(1^\ell)$ and every $m \in \mathcal{M}$,

$$\mathsf{Decrypt}(SK, \mathsf{Encrypt}(PK, m)) = m.$$

One way to define security of a probabilistic public-key encryption scheme is to demand that no adversary can distinguish between the ciphertexts of two known plaintexts. We define this kind of indistinguishability below.

Definition 7.4 (IND-CPA). Let $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a probabilistic public-key public-key encryption scheme. The IND-CPA game proceeds as follows.

Setup The challenger runs $(SK, PK) \leftarrow \mathsf{KeyGen}(1^\ell)$ for some security parameter ℓ, and gives PK along with the description of $\mathcal{M}$ to the adversary.

Query The adversary chooses messages $m_0, m_1 \in \mathcal{M}$ and sends these to the challenger. The challenger chooses a bit $d \in_R \{0, 1\}$, and sends $\mathsf{Encrypt}(PK, m_d)$ to the adversary.

Result The adversary returns his guess d'. It wins if $d' = d$.

We say that an encryption scheme is ϵ-indistinguishable under chosen plaintext attacks, or ϵ-IND-CPA secure for short, if there exists no probabilistic polynomial-time algorithm that can win this game with probability ϵ. If ϵ is negligible in the security parameter ℓ, then we just say the scheme is IND-CPA secure.

The Paillier encryption scheme [Pai99] is implemented as follows.

KeyGen(1^ℓ) Choose two distinct prime numbers p and q such that $n = pq$ is of length ℓ, and such that $\gcd(n, (p-1)(q-1)) = 1$. Compute $\lambda = \mathrm{lcm}(p-1, q-1)$, and randomly select an integer $g \in \mathbb{Z}_{n^2}^*$ such that n divides the order of g (for example, one could take $g = n + 1$, whose order equals n and whose inverse is $n^2 - n + 1$). Return $PK = (n, g)$ and $SK = (p, q, g)$. The message space of the scheme is $\mathcal{M} = \mathbb{Z}_n$.

Encrypt$((n, g), m)$ Choose a random $r \in \mathbb{Z}_n^*$ and return $g^m r^n \bmod n^2$.

Decrypt$((p, q, g), c)$ Compute λ and n from p and q, and return

$$m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n,$$

where[1] $L(u) = \frac{u-1}{n}$.

[1] In the formula defining the Decrypt algorithm the division refers to division modulo n, while in the

Below we will write $c = [\![m]\!]$ if $c \in \mathbb{Z}_{n^2}^*$ is such that $\mathsf{Decrypt}(\lambda, c) = m$. The Paillier encryption scheme is well-suited for our needs, because of the following reasons:

- It is additively homomorphic: that is, given the ciphertexts $[\![m_1]\!] = g^{m_1} r_1^n$, $[\![m_2]\!] = g^{m_2} r_2^n$ of two messages m_1 and m_2, one can calculate $[\![m_1 + m_2]\!]$ by $[\![m_1]\!][\![m_2]\!] = g^{m_1+m_2}(r_1 r_2)^n = [\![m_1 + m_2]\!]$.
- To a verifier that knows a ciphertext $M = [\![m]\!]$ but not m itself, one can prove knowledge of m and the randomness r such that $[\![m]\!] = g^m r^n \bmod n^2$; that is, one can prove

  $$\mathrm{PK}_\Sigma\{(m, r) \colon M = g^m r^n \bmod n^2\},$$

  using the Σ-protocol from (the extended version of) [CDN01]. As this is a Σ-protocol it is only honest-verifier zero-knowledge, but using standard techniques [CDM00] it can be converted to a 4-move zero-knowledge proof of knowledge.

If $n = pq$ with p, q two large primes and z is an integer, then we say that z is a n-residue modulo n^2 if there exists a y such that $z = y^n \bmod n^2$. Pallier proved that his encryption scheme is IND-CPA secure based on the following assumption [Pai99].

Definition 7.5 (Decisional Composite Residuosity Assumption (DCRA)). The DCRA assumption states that there is no probabilistic polynomial-time algorithm that, given a composite n and integer z, can decide whether z is an n-residue modulo n^2 or not, with a probability that is non-negligible in the size of n.

7.2 The Boneh–Boyen signature scheme

There are two versions of the Boneh–Boyen signature scheme [BB08]: a nondeterministic one that is strongly unforgeable under adaptively-chosen message attacks (see Definition 5.23 and the following paragraph on p. 102), and a deterministic one that is weakly unforgeable under chosen-message attacks (as in Definition 7.3). We describe the former first, after which we show how it can be reduced to the latter.

KeyGen(1^ℓ) Generate a Type 3 bilinear group pair (G_1, G_2), such that $|p| = \ell$, with p the order of G_1 and G_2. Pick two generators $P \in G_1$, $Q \in G_2$. Choose two private keys $a, b \in \mathbb{Z}_p^*$, and set $A = aP$ and $B = bP$. The public key is the description of G_1 and G_2, together with (p, e, P, Q, A, B), and the private key is (a, b). The message space is $\mathbb{Z}_p$.

Sign$_{(a,b)}(m)$ Choose a random value $r \in \mathbb{Z}_p \setminus \{-\frac{a+m}{b}\}$ and compute

$$S = \frac{1}{a + m + rb} Q; \tag{7.1}$$

the additions and inverses are calculated modulo p. The signature is the pair (S, r).

definition of L, it refers to integer division. The latter is well-defined, because we have $u^\lambda = 1 \bmod n$ for any $u \in \mathbb{Z}_{n^2}^*$ due to Carmichael's theorem. The demand that n divides the order of g ensures that the inverse of $L(g^\lambda \bmod n^2)$ modulo n always exists.

Verify$_{(p,e,P,Q,A,B)}(m)$ Using the bilinear pairing e, verify that $e(A + mP + rB, S) = e(P, Q)$. Return `true` if and only if this equation holds.

If the signature was correctly generated, then

$$e(A + mP + rB, S) = e\left((a + m + rb)P, \frac{1}{a + m + rb}Q\right) = e(P, Q)$$

so that the signature will verify.

We refer to this scheme as the strong Boneh–Boyen signature scheme. The weak Boneh–Boyen signature scheme is obtained by setting $r = 0$ and removing b and B from the private and public keys, respectively. Thus, a weak Boneh–Boyen signature on the message $m \in \mathbb{Z}_p$ would be $S = \frac{1}{a+m}Q$. The weak scheme is weakly unforgeable under chosen-message attacks in the sense of Definition 7.3. Boneh and Boyen prove the unforgeability of their strong scheme by reducing it to that of their weak scheme. The unforgeability of the weak scheme, in turn, relies on the Strong Diffie-Hellman assumption.

Definition 7.6 (Strong Diffie-Hellman assumption). The q-Strong Diffie-Hellman assumption holds in G_1, G_2 if, when given as input a $(q + 3)$-tuple of elements $(P, xP, Q, xQ, x^2Q, \ldots, x^qQ) \in G_1^2 \times G_2^{q+1}$, it is intractable to output a pair $\left(d, \frac{1}{x+d}Q\right) \in \mathbb{Z}_p \times G_2$ for a freely chosen value $d \in \mathbb{Z}_p \setminus \{-x\}$.

The Strong Diffie-Hellman (SDH) assumption is the unparameterized version of this assumption, as follows. Let $e_\alpha \colon G_{1,\alpha} \times G_{2,\alpha} \to G_{T,\alpha}$ be a family of pairings, such that $G_{1,\alpha}$ and $G_{2,\alpha}$ are group families, as in Section 5.2. Then the SDH assumption holds in $\{G_{1,\alpha}, G_{2,\alpha}\}_\alpha$ if for any positive polynomial $q \colon \mathbb{Z} \to \mathbb{N}$, the $q(|\alpha|)$-Strong Diffie-Hellman assumption holds within $G_{1,\alpha}$ and $G_{2,\alpha}$.

Like the whLRSW and (X)KEA assumptions from Chapter 9, the SDH assumption can be proven to hold in the generic group model (see Section 9.5.2). For a more elaborate description of the assumption, and of the signature scheme and its properties, we refer to the original paper by Boneh and Boyen [BB08].

7.3 The partially blind Boneh–Boyen scheme

To include the common information $k \in \mathcal{I}$, we use the following special case of the Generalized Strong Boneh–Boyen (GSBB) signature scheme [Bha+09]. We take the strong Boneh–Boyen signature scheme and include $c \in \mathbb{Z}_p^*$ and $C = cP$ in the private and public keys, respectively, i.e.,

$$SK = (a, b, c) \quad \text{and} \quad PK = (p, e, P, Q, A, B, C).$$

The message space $\mathcal{M}$ and common-information space $\mathcal{I}$ both are $\mathbb{Z}_p$. For $m, k \in \mathbb{Z}_p$, a signature will be $(S, r) \in G_2 \times \mathbb{Z}_p$, where

$$S = \frac{1}{a + m + rb + kc}Q.$$

Common information: Boneh–Boyen public key (e, p, P, Q, A, B, C), common information $k \in \mathcal{I}$

User		**Signer**
knows message $m \in \mathbb{Z}_p$		knows secret keys $a, b, c \in \mathbb{Z}_p$
	Phase 1	
Generate Paillier (p', q', g)		
Choose $\beta, r_1 \in_R \mathbb{Z}_p^*$		
send $n = p'q'$, g, $[\![\beta]\!]$, $[\![\beta r_1]\!]$	$\longrightarrow$	into n, g, X, Y
		$\mathrm{PK}\{(\beta, \delta) : X = [\![\beta]\!] \wedge Y = [\![\delta]\!]\}$
	Phase 2	
		choose $r_2 \in_R \mathbb{Z}_p$
		choose $\gamma \in_R \{0, \ldots, n - 2p^2 - 3p^3\}$
into D	$\longleftarrow$	send $X^{a+r_2 b+kc} Y^b [\![\gamma]\!]$
		$\mathrm{PK}_\Sigma\{(\rho, b, \gamma) : D = X^\rho Y^b [\![\gamma]\!] \bmod n^2\}$
	Phase 3	
set $\tilde{s} \leftarrow \mathsf{Decrypt}(D)$		
send $\tilde{s} + \beta m \bmod p$	$\longrightarrow$	into $\hat{s}$
	Phase 4	
		set $s = \hat{s} - (\gamma \bmod p)$
into $\bar{S}, r_2$	$\longleftarrow$	send $\frac{1}{s} Q, r_2$
	Phase 5	
set $S = \beta \bar{S}$, $r = r_1 + r_2$		
$\mathsf{Verify}_{PK}(m, k, S)$		
return (S, r)		

Figure 7.1. Our interactive partially blind signing protocol from Definition 7.7.

The signature is valid only if

$$e(A + mP + rB + kC, S) = e(P, Q).$$

Definition 7.7 (Partially blind signing protocol). To obtain a signature on a message m and common information k (on which the user and signer agreed in advance), the user and the signer (who knows the private keys a, b, c), interact in the following way[2] (see also Figure 7.1).

1. The user generates a new Paillier encryption system (p', q', g) (we use accents to prevent a notational clash with the modulus p of $\mathbb{Z}_p$) such that $n = p'q'$ exceeds p^4, and such that factoring n is infeasible. Then, it generates a blinding factor $\beta \in_R \mathbb{Z}_p^*$ and a randomizer $r_1 \in_R \mathbb{Z}_p^*$ for use in the resulting signature, and sets $X = [\![\beta]\!]$ and

[2]The private keys a, b, c of the signer, as well as the messages m, k and randomness r_1, r_2, of a signature are all elements of $\mathbb{Z}_p$. In the protocol we embed these elements in the message space of the Paillier encryption scheme, which is $\mathbb{Z}_n$ (for some n that we will take to be much larger than p). This means that when adding and multiplying these numbers in the protocol below, we refer to integer arithmetic (as opposed to arithmetic within $\mathbb{Z}_p$). In the third step of the protocol, there is a reduction modulo p that restores them as elements of $\mathbb{Z}_p$.

$Y = [\![\beta r_1]\!]$. It sends n, g, X, Y to the signer, and proves that it knows the plaintexts of X and Y:

$$\mathrm{PK}\{(\beta, \delta) : X = [\![\beta]\!] \ \wedge \ Y = [\![\delta]\!]\}.$$

2. If the proof is correct the signer generates randomness $r_2 \in_R \mathbb{Z}_p$ and a blinding term $\gamma \in_R \{0, \ldots, n - 2p^2 - 3p^3\}$ (this maximum value for γ ensures that no reduction modulo n will occur). The signer calculates

$$D = X^{a+r_2 b+kc} Y^b [\![\gamma]\!] \bmod n^2 = [\![\beta(a + (r_1 + r_2)b + kc) + \gamma \bmod n]\!].$$

It sends D to the user and proves that it constructed D correctly using a Σ-protocol:

$$\mathrm{PK}_\Sigma\{(\rho, b, \gamma) : D = X^\rho Y^b [\![\gamma]\!] \bmod n^2\}.$$

3. The user calculates

$$\tilde{s} = \mathsf{Decrypt}(D) + \beta m \bmod n = \beta(a + m + (r_1 + r_2)b + kc) + \gamma \bmod n,$$

and sends $\hat{s} = \tilde{s} \bmod p$ to the signer (here, the reduction modulo p is done by taking the smallest positive representative of $\tilde{s}$).
4. The signer removes γ by calculating $s = \hat{s} - \gamma \bmod p$, and sends $\bar{S} = \frac{1}{s} Q$ together with r_2 to the user.
5. Finally, the user unblinds the signature to obtain

$$S = \beta \bar{S} = \frac{1}{a + m + (r_1 + r_2)b + kc} Q.$$

Setting $r = r_1 + r_2$, it accepts if (S, r) is a valid signature on m and k.

The protocol is summarized in Figure 7.1. It consists of 9 moves. In the security proofs we will need a simulator (against possibly dishonest verifiers) as well as an extractor for the proof of knowledge performed by the user in step 1, so that it must be a zero-knowledge proof of knowledge. However, for the proof of knowledge from step 2 performed by the signer we need no such simulator but only an extractor, so that it suffices for this proof to be a Σ-protocol.

Proposition 7.8. *The blind Boneh–Boyen signature scheme as described in Definition 7.7 is correct.*

Proof. If both the user and the signer follow the protocol, then at the start of step 3 the user calculates the value

$$\tilde{s} = \beta\rho + \beta r_1 b + \gamma + \beta m \bmod n = \beta(a + m + (r_1 + r_2)b + kc) + \gamma \bmod n. \qquad (7.2)$$

It is clear that this results in a valid signature if no modular reduction occurred, i.e., if $\tilde{s} = \beta(a + m + (r_1 + r_2)b + kc) + \gamma$. We now show that this always the case. Consider the expression $\beta(a + m + (r_1 + r_2)b + kc)$. The maximum value of each of these variables is

at most $p-1$, hence $\beta(a+m+(r_1+r_2)b+kc) < p(p+p+(p+p)p+p^2) = 2p^2+3p^3$. Since γ is chosen from $\{0,\ldots,n-2p^2-3p^3\}$, it follows that $\tilde{s}$ indeed equals $\beta(a+m+(r_1+r_2)b+kc)+\gamma$ and that no modular reduction occurred. □

The purpose of the number γ is to make the number $\tilde{s} = d+\gamma := \beta(a+(r_1+r_2)b+kc)+\gamma$ appear randomly distributed in $\mathbb{Z}_n$, so that it leaks no information about a, b and c. This will be the case unless $d+\gamma < 2p^2+3p^3$ or $d+\gamma > n-2p^2-3p^3$. Denote with E the event that either of those two happens. Recalling that $n > p^4$ and that $p > 2^{\ell-1}$ (where ℓ is the security parameter), we have

$$P(E) = \frac{2(2p^2+3p^3)}{n} < \frac{8p^3}{n} < \frac{8p^3}{p^4} = \frac{8}{p} < 2^{-\ell+4},$$

which is negligible in ℓ.

7.3.1 Blind Boneh–Boyen signatures

If we include no common information by setting $k=0$ and removing c and C from the signer's private and public key, then Figure 7.1 reduces to a blind signing protocol for Boneh–Boyen signatures, and the theorems below then prove that this protocol is blind and that the scheme is unforgeable.

7.4 Blindness and unforgeability

In this section we prove that the signer learns nothing about the message and signature (blindness), and that an attacker cannot forge signatures even if it can use the interactive signing protocol at will.

7.4.1 Blindness

We reduce the partial blindness of our signing protocol to the IND-CPA indistinguishability (as in Definition 7.4) of the Paillier encryption scheme. The idea of the proof hinges on two facts. One, the adversary learns nothing about the values of β and βr_1 because they are Paillier encrypted. Two, the remaining values in the adversary's view can correspond to any signature. We prove the former for β by arguing that the advantage of an adversarial signer cannot depend on the plaintext of X that it receives in the first step of the signing protocol (see Definition 7.7). A similar argument then shows the same for βr_1.

Definition 7.9. We say that a user sends the correct X in the first step of the signing protocol when the plaintext β of X is the same β that occurs in the value $\hat{s} = \beta(a+m+rb+kc)+\gamma \bmod p$ that the user sends in step 3.

Lemma 7.10. *Consider an adversarial signing algorithm whose advantage for winning the blindness game (as in Definition 7.1) for our partially blind Boneh–Boyen signing algorithm is* ϵ_+

when both simulated users $\mathcal{U}_0$ and $\mathcal{U}_1$ send the correct X, and ϵ_- when both users do not send the correct X. If Paillier is ϵ-IND-CPA secure, then $|\epsilon_+ - \epsilon_-|/2 < \epsilon$.

Proof. Suppose that we have an adversary $\mathcal{A}$ whose advantage at the blindness game is ϵ_+ when the users send the correct X, and ϵ_- when the users do not. We assume[3] $\epsilon_+ > \epsilon_-$, and set $(\epsilon_+ - \epsilon_-)/2 = \epsilon'$. We build an algorithm $\mathcal{B}$ that has an advantage ϵ' at winning the IND-CPA game. $\mathcal{B}$ will act as the adversary in the IND-CPA game, and as the challenger in the blindness game. It proceeds as follows.

1. The challenger of $\mathcal{B}$ outputs a Paillier public key PK. $\mathcal{B}$ chooses $\beta_0, \beta_1 \in \mathbb{Z}_p$ and sends these to his challenger, who responds by choosing $d \in_R \{0,1\}$ and sending $[\![\beta_d]\!]$ to $\mathcal{B}$.
2. The adversary $\mathcal{A}$ generates a partially blind Boneh–Boyen public-private keypair, common information $k \in \mathcal{I}$ and messages $m_0, m_1 \in \mathcal{M}$. It sends the public key and k, m_0, m_1 to $\mathcal{B}$.
3. Distinguisher $\mathcal{B}$ flips a bit $b \in_R \{0,1\}$, and gives m_b to simulated user $\mathcal{U}_0$ and $m_{\bar{b}}$ to simulated user $\mathcal{U}_1$. The users engage in the signing protocol with the adversary $\mathcal{A}$. In step 1, user $\mathcal{U}_0$ sends $X = [\![\beta_d]\!]$ while user $\mathcal{U}_1$ sends $X = [\![\beta_d]\!]^{\beta_1 - \beta_0} = [\![\beta_d + \beta_1 - \beta_0]\!]$. Note that if $d = 0$ then they both send the correct X, while if $d = 1$ then both of them do not.
   The users calculate Y honestly, i.e., $Y = [\![\beta_i r_1]\!]$ for user $\mathcal{U}_i$. They simulate the zero-knowledge proof of step 1.
4. At the end of step 2, both users extract ρ, b, γ from the zero-knowledge proof performed by $\mathcal{A}$. Then, user $\mathcal{U}_0$ calculates

   $$\hat{s} = \big(\beta_0(\rho + m_b + r_1 b) + \gamma\big) \bmod p$$

   and similarly for user $\mathcal{U}_1$. They send this value for $\hat{s}$ to $\mathcal{A}$ in step 3. Thus, the value $\hat{s}$ is what it would normally be (c.f. equation (7.2)), while X is either correct for both users if the challenger chose $d = 0$, or incorrect for both users if $d = 1$.
5. Challenger $\mathcal{A}$ outputs a guess b'. Distinguisher $\mathcal{B}$ outputs $d' = 0$ if $\mathcal{A}$ wins (i.e., $b = b'$) and $d' = 1$ if it loses.

We now calculate the probability that $\mathcal{B}$ outputs the correct answer. If $d = 0$ then $\mathcal{B}$ outputs the correct guess $d' = 0$ if $\mathcal{A}$ wins, which happens with probability $\frac{1}{2} + \epsilon_+$. If $d = 1$, then $\mathcal{B}$ outputs the correct guess $d' = 1$ if $\mathcal{A}$ loses, which happens with probability $\frac{1}{2} - \epsilon_-$. The chance of $\mathcal{B}$ winning is thus

$$\frac{1}{2}\left(\frac{1}{2} + \epsilon_+\right) + \frac{1}{2}\left(\frac{1}{2} - \epsilon_-\right) = \frac{1}{2} + \frac{1}{2}(\epsilon_+ - \epsilon_-),$$

meaning that the advantage of $\mathcal{B}$ at winning the IND-CPA game is $(\epsilon_+ - \epsilon_-)/2 = \epsilon'$. Thus, if Paillier encryption is ϵ-IND-CPA secure, it follows that indeed $\epsilon' = (\epsilon_+ - \epsilon_-)/2 < \epsilon$. □

[3] The opposite assumption does not make much sense, but with minor modifications the proof can then still be made to work.

With small modifications in the reduction above, it is easy to prove a similar statement where not both but just one of the users sends a correct X. Furthermore, it is clear that the same arguments can be applied to $Y = [\![\beta r_1]\!]$.

Theorem 7.11. *No adversary can break the partial blindness of the partially blind Boneh–Boyen signature scheme with advantage ϵ, provided that Paillier is $(\epsilon/2)$-IND-CPA secure.*

Proof. The view of the signer of a signing is

$$D = \{k, [\![\beta]\!], [\![\beta r_1]\!], \beta(\rho + r_1 b' + m) + \gamma \bmod p, \gamma, r_2\}.$$

We use a prime on the number b' because it is nowhere enforced that it equals the signer's secret key b (we will discuss this subtlety shortly). Let us call the second, third and fourth elements of this trace X, Y and $\hat{s}$ respectively, so that $D = \{k, X, Y, \hat{s}, \gamma, r_2\}$. By the lemma above, the adversary cannot use X or Y to his advantage, so we can leave them out. In addition, the common information k has the same value, independent from the choice for d that the challenger makes.

Therefore the only pieces of information that could possibly increase the adversary's advantage are the values $\hat{s}, \gamma$ and r_2. Now as it is nowhere enforced in the signing protocol that $b' = b$, we must consider the possibility that the adversary chooses $b' = 0 \bmod p$. But then r_1 vanishes from $\hat{s}$, and we have already shown that the adversary cannot use $Y = [\![\beta r_1]\!]$, the only other element of its view that depends on r_1, to its advantage. It follows that the value for r_1 is completely hidden from the adversary, so that it has no way of knowing which value for r_2 would result in a valid signature. Thus the output in this case is with overwhelming probability an invalid signature, so that the adversary automatically loses.

We may therefore restrict our attention to the possibility that $b' \neq 0 \bmod p$. In that case, however, for any valid tuple (m, k, r, S) there exist numbers $\beta, r_1 \in \mathbb{Z}_p^*$ such that $r = r_1 + r_2$ and $\hat{s} = \beta(\rho + r_1 b' + m) + \gamma \bmod p$ (even if $b' \neq b \bmod p$). Thus, any view can correspond to any message-signature pair. □

7.4.2 Unforgeability

Theorem 7.12. *If the weak Boneh–Boyen scheme is (q, ϵ)-weakly unforgeable under chosen-message attacks, then our Blind Boneh–Boyen scheme is (q, ϵ')-unforgeable under chosen-message attacks, where $\epsilon' - \epsilon$ is negligible.*

Proof. Suppose we have an adversary $\mathcal{A}$ that breaks the unforgeability of our scheme under chosen-message attacks. In order to prove the theorem, we will need to create not one but three challengers $\mathcal{B}_A$, $\mathcal{B}_B$, $\mathcal{B}_C$ for $\mathcal{A}$. Our argument will run as follows: challenger $\mathcal{B}_A$ will either forge weak Boneh–Boyen signatures, or it will fail. But if it succeeds, then we can use either challenger $\mathcal{B}_B$ or $\mathcal{B}_C$ to solve discrete logarithms. Since none of these things can happen, it follows that our scheme must be unforgeable.

In the proof below, the adversary $\mathcal{A}$ plays the unforgeability game for partially blind signature schemes with algorithms $\mathcal{B}_A$, $\mathcal{B}_B$ and $\mathcal{B}_C$, who on the one hand serve as challenger for $\mathcal{A}$. On the other hand, these algorithms simultaneously act as the adversary in the unforgeability game for weak Boneh-Boyen signatures; we will construct them such that they use what they learn from adversary $\mathcal{A}$ to win this game. They act as follows.

Setup Challengers $\mathcal{B}_A$, $\mathcal{B}_B$ and $\mathcal{B}_C$ choose q messages $w_1, \ldots, w_q \in \mathbb{Z}_p$ and send them to their own challenger from the weak Boneh–Boyen unforgeability game as the messages that it wants signed. In response, the weak Boneh–Boyen challenger chooses a private-public key pair, that we will denote with (a, A), (b, B) and (c, C) for challengers $\mathcal{B}_A$, $\mathcal{B}_B$, $\mathcal{B}_C$ respectively. The weak Boneh–Boyen challenger signs the messages $w_1, \ldots, w_q$, resulting into signatures $S_1, \ldots, S_q$, and sends these to our challengers.
Next, our challengers create the two missing private-public key pairs (for example, $\mathcal{B}_B$ chooses $a, c \in_R \mathbb{Z}_p^*$, and sets $A = aP, C = cP$). They then send A, B, C to the adversary $\mathcal{A}$.

Queries Proceeding adaptively, adversary $\mathcal{A}$ engages with our challengers in the blind Boneh–Boyen signing algorithm for q message-pairs $(m_1, k_1), \ldots, (m_q, k_q)$. Our three challengers all perform the following actions on the j-th query.

- At the end of step 1, our challenger extracts the values β, δ that $\mathcal{A}$ chose for message m_j from the zero-knowledge proof, and sets $r_1 = \delta/\beta$.
- In step 2, $\mathcal{B}_A$ chooses $a' \in_R \mathbb{Z}_p$ and acts as if this is the private key of his challenger; similarly, $\mathcal{B}_B$ fakes b and $\mathcal{B}_C$ fakes c. In addition, they act as if $r_2 = 0$. It runs the zero-knowledge proof normally. As D is blinded by γ and the zero-knowledge proof is hiding, $\mathcal{A}$ cannot detect these changes.
- In step 3, $\mathcal{B}_A$ learns $\beta(a' + m_j + r_1 b + k_j c)$. As it knows all variables except for m_j, it can solve this to learn m_j. Challengers $\mathcal{B}_B$ and $\mathcal{B}_C$ can learn the message m_j in much the same way.

Our challengers now answer the adversary's query as follows.

Challenger $\mathcal{B}_A$ chooses $r_2 \in \mathbb{Z}_p$ such that $m_j + (r_1 + r_2)b + k_j c = w_j \bmod p$. For this message it received a valid signature S_j in the announcement phase. Then, it sends $(\beta^{-1} S_j, r_2)$ back to $\mathcal{A}$ in step 4.

Challenger $\mathcal{B}_B$ sets $r_2 = (a + m_j + k_j c)/w_j - r_1$ and $\bar{S} = w_j/(a + m_j + k_j c) S_j$, and sends $(\beta^{-1}\bar{S}, r_2)$ to $\mathcal{A}$ in step 4. From the point of view of the adversary this indeed results into a valid signature over its query:

$$
\begin{aligned}
& e(A + mP + rB + kC, \bar{S}) \\
&\quad = e\Big(\Big(a + m_j + b\frac{m_j + a + k_j c}{w_j} + k_j c\Big)P, \frac{w_j}{a + m_j + k_j c} S_j\Big) \\
&\quad = e\Big(\Big((b + w_j)\frac{m_j + a + k_j c}{w_j}\Big)P, \frac{w_j}{a + m_j + k_j c}\frac{1}{b + w_j} Q\Big) \\
&\quad = e(P, Q).
\end{aligned}
$$

Challenger $\mathcal{B}_C$ sets $r_2 = (kw_j - m - a)/b - r_1$ and $\bar{S} = k^{-1}S_j$, and sends $(\beta^{-1}\bar{S}, r_2)$ to $\mathcal{A}$ in step 4. From the point of view of the adversary this indeed results into a valid signature over its query:

$$\begin{aligned} & e(A + mP + rB + kC, \bar{S}) \\ &= e\Big(\Big(a + m + \frac{kw_j - m - a}{b}b + kc\Big)P, \frac{1}{k}S_j\Big) \\ &= e\Big((kw_j + kc)P, \frac{1}{k}\frac{1}{c + w_j}Q\Big) = e(P, Q). \end{aligned}$$

Output For reasons that will become clear shortly, it suffices here to consider only challenger $\mathcal{B}_A$. If adversary $\mathcal{A}$ sends $k \in \mathcal{I}$ and $q_k + 1$ tuples (m_j, k, r_j, S_j) to our challengers, and if one of the tuples from the adversary is such that $m_j + r_j b + kc \neq w_h$ for all $1 \leq h \leq q$, then $\mathcal{B}_A$ sends $(m_j + r_j b + kc, S_j)$ to his challenger, otherwise it aborts.

The output of $\mathcal{B}_A$ will be correct if and only if that of $\mathcal{A}$ is correct. Thus it only remains to show that there is a j such that $m_j + r_j b + kc \neq w_h$ for all k. We now show that this is so with overwhelming probability.

Indeed, suppose that adversary $\mathcal{A}$ won the game. This implies that all $q_k + 1$ tuples (m_j, r_j, S_j) are distinct. Since (m_j, k, r_j) uniquely determines S_j, this means that all $q_k + 1$ pairs (m_j, r_j) must be distinct. But now suppose that there are two unequal numbers j, k such that

$$m_j + r_j b + kc = w_k = m_h + r_h b + k_h c \bmod p. \tag{7.3}$$

This means that b or c could be calculated by

$$b = \frac{m_j - m_h + c(k_j - k_h)}{r_h - r_j} \bmod p, \qquad c = \frac{m_j - m_h + b(r_j - r_h)}{k_h - k} \bmod p.$$

Notice that at least one of these will work:

- If $k_h \neq k$ then the right equation will result in c.
- If $k_h = k$ then $r_k \neq r_j$, otherwise not all pairs (m_j, r_j) would be distinct by equation (7.3), so the left equation will give b.

By using the challengers $\mathcal{B}_B$ or $\mathcal{B}_C$ constructed above, so that we do not need to know b or c in advance, this leads to a violation of the discrete logarithm problem. Additionally, since $\mathcal{B}_B$ and $\mathcal{B}_C$ now know the private key of their weak Boneh–Boyen challenger, they can easily win the weak Boneh–Boyen unforgeability game.

We conclude that there must with overwhelming probability be a j such that $m_j + r_j b + kc \neq w_h$. This means that in this case the output $(m_j + r_j b + kc, S_j)$ of challenger $\mathcal{B}_A$ is new (i.e., unequal to all message-signature pairs that it received in the Setup phase), so that $\mathcal{B}_A$ wins the weak Boneh–Boyen unforgeability game. Therefore, if $\mathcal{A}$ has advantage ϵ, then the advantage of $\mathcal{B}_A$ will be negligibly close to ϵ. □

7.5 Attribute-based credentials using our scheme

In this section we show that we can use our new blind signature scheme to construct a (single-show) attribute-based credential scheme that is similar to U-Prove. Our scheme improves U-Prove by not relying on the random oracle model, and by being provably unforgeable. Such an unforgeabilty proof is not known for U-Prove credentials; in fact, it has even been suggested that no such proof exists under any standard intractability assumption [BL13b]. We first explain how to extend our signatures to credentials by including multiple attributes, after which we show how these attributes can be selectively disclosed.

7.5.1 Signatures as credentials

In our credential scheme our partially blind Boneh–Boyen signatures become credentials. The attributes are represented by several pieces of common information. To encode these attributes, we replace the signer's secret key $c \in \mathbb{Z}_p^*$ by a tuple of numbers[4] $(c_1, \ldots, c_n) \in (\mathbb{Z}_p^*)^n$ (and the public key C by $C_1 = c_1 P, \ldots, C_n = c_n P$). Then the signer can sign a hidden message m and a tuple $k_1, \ldots, k_n$ by setting

$$D = X^{\rho} Y^{b} [\![\gamma]\!] \qquad \text{with} \qquad \rho = a + r_2 b + \sum_{i=1}^{n} k_i c_i$$

in step 2 of the Sign protocol (see p. 132). The resulting signature (S, r) over $(m, k_1, \ldots, k_n)$ is then valid only if $e\big(A + mP + rB + \sum_{i=1}^{n} k_i C_i, S\big) = e(P, Q)$.

This results in a partially blind signature scheme for GSBB signatures [Bha+09]. This scheme reduces to the scheme from the previous sections for $n = 1$. It is not difficult to adapt the unforgeability and blindness proofs to this scheme.

To issue a credential, the user and the issuer decide in advance on the attributes $k_1, \ldots, k_n$ that the credential will have. Then, the user runs the Sign protocol with the issuer on $(m, k_1, \ldots, k_n)$, where the user chooses $m \in_R \mathbb{Z}_p$. The user receives a signature (S, r) on the attributes $(k_1, \ldots, k_n)$ and message m. The credential then consists of the message, the attributes and the signature. The partial blindness of the scheme ensures that the issuer does not learn m, nor can it later recognize the resulting signature.

7.5.2 Showing a credential

The user can show such a credential as follows. Let $\mathcal{D} \subset \{1, \ldots, n\}$ be the index set of the attributes that the user wants to disclose, and let $\mathcal{C} = \{1, \ldots, n\} \setminus \mathcal{D}$ be the remaining hidden attributes.

- The user sends S, r and the attributes $(k_i)_{i \in \mathcal{D}}$ that it wishes to disclose to the verifier, together with $D = mP + \sum_{i \in \mathcal{C}} k_i C_i$.

[4] In the remainder of this chapter we will reuse the letter n to denote the number of attributes, for consistency with the next chapters.

- The user performs a zero-knowledge proof of knowledge of the message m and the hidden attributes:

$$\mathrm{PK}\left\{(m, (k_i)_{i\in\mathcal{C}}) \colon D = mP + \sum_{i\in\mathcal{C}} k_i C_i\right\}. \tag{7.4}$$

- The verifier checks that the signature (S, r) is valid as follows:

$$e\left(A + rB + D + \sum_{i\in\mathcal{D}} k_i C_i, S\right) \stackrel{?}{=} e(P, Q).$$

Theorem 7.12 (or rather its generalization to multiple pieces of common information, as mentioned in the previous subsection) then guarantees unforgeability of these credentials. In addition, as a consequence of Theorem 7.11, the scheme offers issuer unlinkability (see Section 5.7.3 on p. 107).

The signatures (S, r) of our scheme are about the same size as those of U-Prove in the ECC setting. If we use standard Σ-protocols like U-Prove, then the showing protocols are approximately equally efficient. Furthermore, contrary to U-Prove the proof of the scheme does not rely on the random oracle model. Finally, if we do not want to assume that the verifier is honest in the Σ-protocols then we can easily use a black-box zero-knowledge proof of knowledge instead, such as for example the one from Example 5.21, or the Schnorr Σ-protocol in the Fiat-Shamir heuristic.

Remark 7.13. It is interesting to note how the partially blind signature scheme resulted in an attribute-based credential scheme almost for free. Broadly speaking, we only used the following features of our scheme:

- It allows multiple pieces of common information k_i, which can then serve as the attributes of our credentials;
- The structure of the signatures allows the hiding of some of the contained attributes through a proof of knowledge as in equation (7.4).

Other partially blind signature schemes that offer similar features may thus also result in interesting attribute-based schemes in this fashion; it would be interesting to see if one of them might also support unlinkability.

At the same time, it is important to note that this is definitely not the only route to unlinkable attribute-based schemes. Indeed, in Chapter 9 we create an unlinkable scheme whose Issue protocol (see p. 160) is not partially blind; instead, the unlinkability will come from the ShowCredential protocol. We will return to this difference in Remark 9.17 on p. 167.

7.6 Related work

Introduced by Chaum [Cha83], blind signature schemes were formally defined by Juels, Luby and Ostrovsky [JLO97]. Since then much work on the subject has been done, both

in the random oracle and standard models; for example, see [Bel+02; Bol02; CKW05; Poi98].

Partially blind signature schemes were introduced by Abe and Fujisaki [AF96] and further formalized by Abe and Okamoto [AO00]. The schemes of both of these papers use the random oracle model, as well as those from [Cho+05; ZSS03]. A newer scheme by Okamoto [Oka06] is secure in the standard model like our scheme. The issuing protocol of this scheme is more efficient than ours, consisting of 4 moves (compared to 9); on the other hand, his scheme uses a Type 2 pairing (i.e., there must exist an efficiently computable homomorphism from G_2 to G_1, see Section 5.4); our scheme uses Type 3 pairings which are generally more efficient [GPS08]. Moreover, Okamoto bases the unforgeability of his scheme on a nonstandard hardness assumption.

Belenkiy et al. [Bel+09] created a two-party computation protocol for computing weak Boneh–Boyen signatures, that bears some resemblance to our issuing protocol. On the one hand their protocol is slightly more efficient than ours; specifically, the message space of the encryption scheme is smaller due to the simpler structure of the signatures that are computed. On the other hand, because weak Boneh–Boyen signatures are deterministic their protocol is unsuitable for (partially) blind signatures as well as an application to attribute-based credentials as in Section 7.5.

A lesser attractive feature of our issuing protocol is that it consists of 9 moves, which is far from optimal. We believe that this is because of the structure of Boneh–Boyen signatures, in particular the modular inversion in equation (7.1), and indeed all schemes that we know of that issue Boneh–Boyen like signatures have a similar amount of moves. There are, however, signature schemes that allow issuing protocols of fewer moves: for example, [Bla+13] introduces a blind issuing protocol for Waters signatures [Wat05] of just two moves, and [FHS15] introduces a partially blind signature scheme in the standard model, also of 2 moves.

As already mentioned, our attribute-based credential scheme from Section 7.5 improves on U-Prove by being provably secure and not relying on the random oracle model. We would also like to point out the single-show attribute-based credential scheme by Baldimtsi and Lysyanskaya [BL13a], which is provably secure like ours, but requires the random oracle model like U-Prove.

Finally, Weitenberg's Master's thesis [Wei12] also contains an interactive issuing protocol for Boneh–Boyen like signatures, as well as an application to attribute-based credentials. There are some differences between his and our signing protocol, however; he did not include a convincing unforgeability proof; and while an unlinkable scheme was aimed at, Weitenberg was not able to prove unlinkability.

7.7 Conclusion

We introduced a partially blind signing protocol for Boneh–Boyen-like signatures and proved its security. This enables applications of these signatures in situations where it is important that the issuer does not learn (part of) the message being signed nor the resulting signature. As an example of the simplicity and flexibility of our scheme, we introduced a single-show attribute-based credential scheme that improves on its

well-known predecessor U-Prove by being provably unforgeable without the use of the random oracle model, using only standard hardness assumptions. The length of these credentials is independent of the number of attributes, resulting in a very efficient ShowCredential protocol.

In the next chapter, we show how credentials from the scheme from [HK14], in which the length of the credentials is also constant, can be forged. Finally, Chapter 9 will introduce a new scheme which is provably unforgeable and unlinkable, in which the credential size is linear in n.

Chapter 8

The self-blindable U-Prove scheme from FC'14 is forgeable

Recently an unlinkable version of the U-Prove attribute-based credential scheme was proposed at Financial Crypto '14 [HK14]. Unfortunately, the new scheme is forgeable: if sufficiently many users work together then they can construct new credentials, containing any set of attributes of their choice, without any involvement of the issuer. In this chapter we show how they can achieve this, and we point out the error in the unforgeability proof.

This chapter is based on the following article.

[VRH16] E. Verheul, S. Ringers, and J.-H. Hoepman. "The self-blindable U-Prove scheme from FC'14 is forgeable". In: *Financial Cryptography and Data Security – FC'16* (2016). In print. URL: https://eprint.iacr.org/2015/725.

The main idea of the attack that we describe in this chapter and the corresponding article is due to Eric Verheul. The specifics of applying the attack to Hanzlink and Kluczniak's scheme is my own work, as well as the article and the text below.

8.1 Introduction

In all attribute-based credential schemes that we considered so far (notably the one from Chapter 7 and U-Prove, both of which are defined in prime-order elliptic curves), the length of the credentials does not depend on the amount of attributes n. In both of these cases, this resulted in short signatures and efficient ShowCredential protocols, that however do not offer multi-show unlinkability.

In an attempt to fix this lack of unlinkability while retaining efficiency, L. Hanzlik and K. Kluczniak proposed in [HK14] a new scheme that is based on U-Prove but uses a different, self-blindable signature scheme, based on the self-blindable construction by Verheul [Ver01]. Although [HK14] does contain an argument for the unforgeability of their scheme, we show here that this argument contains an error, and that the proposed construction is forgeable, in the sense that if sufficiently many users collude then they can construct new credentials containing arbitrary attributes of their choice, without involvement of the issuer.

8.2 The credential scheme

Hanzlik and Kluczniak [HK14] present their blindable U-Prove scheme as an extension of the original U-Prove scheme, in the following sense: a self-blindable signature (based on [Ver01]) is added to a U-Prove credential. When showing a credential, the user can then choose to either show his credential using the original linkable U-Prove ShowCredential protocol, or using a new protocol that uses the new self-blindable signature and should offer unlinkability. Since we are concerned only with the forgeability of the self-blindable construction, our description of the credential scheme will omit details that are relevant only to the original construction.

The setup is as follows.[1] q is a prime number of length k, and $e\colon G_1 \times G_2 \to G_T$ is a bilinear pairing of Type 2 (see Section 5.4 on p. 91), where q is the order of G_1, G_2 and G_T. The issuer's public key is

$$(q, e, g_0, \ldots, g_n, p, p', p_0, p_1),$$

where

- $g_0, \ldots, g_n$ are random generators of G_1,
- p and p' are random generators of G_2,
- $p_0 = (p')^z$,
- $p_1 = p^f$.

The tuple $(f, z) \in \mathbb{Z}_q^2$ is the issuer's secret key.

A credential consists of the tuple

$$((x_1, \ldots, x_n), (h, h_2, h_3, h_4, \alpha, b_1, b_2))$$

where

- $x_1, \ldots, x_n \in \mathbb{Z}_q$ are the attributes,
- $\alpha, b_1, b_2 \in \mathbb{Z}_q$, chosen by the user during issuing of the credential,
- $h = (g_0 g_1^{x_1} \cdots g_n^{x_n})^\alpha$,
- $h_2 = h^f$,
- $h_3 = h^{b_1} h_2^{b_2}$,

[1] In this chapter we use the notations of the article by Hanzlik and Kluczniak, meaning that we deviate from much of our earlier notational conventions.

- $h_4 = h_3^z = (h^{b_1} h_2^{b_2})^z$.

The validity of the credential can be checked by

$$e(h, p_1) \stackrel{?}{=} e(h_2, p) \qquad \text{and} \qquad e(h_3, p_0) \stackrel{?}{=} e(h_4, p').$$

Such a credential can be blinded into a new one as follows. Take random $k, \ell \in \mathbb{Z}_q^*$, and set $(\bar{h}, \bar{h}_2, \bar{h}_3, \bar{h}_4) = (h^k, h_2^k, h_3^{k\ell}, h_4^{k\ell})$. Then

$$((x_1, \ldots, x_n), (\bar{h}, \bar{h}_2, \bar{h}_3, \bar{h}_4, \alpha k, b_1 \ell, b_2 \ell))$$

is a new, valid credential over the same attributes. In [HK14] a ShowCredential protocol for these credential is provided, in which the credentials are blinded as above. The protocol should offer unlinkability but it is not proven that it does (and we have not checked this).

8.3 Forging new credentials

8.3.1 Constructing signatures on the elements g_i

We first show that if sufficiently many users work together, then for each i they can compute a tuple g_i^f, g_i^z, g_i^{fz}, even though f and z are private to the issuer. Using these tuples they can easily create new valid credentials over any set of attributes of their choice. Since this will involve many credentials, we will write the elements from the credential of user j with an extra subscript j:

$$((x_{1,j}, \ldots, x_{n,j}), (h_j, h_{2,j}, h_{3,j}, h_{4,j}, \alpha_j, b_{1,j}, b_{2,j})).$$

The element h_j is of the form

$$h_j = (g_0 g_1^{x_{1,j}} \cdots g_n^{x_{n,j}})^{\alpha_j}.$$

By blinding the credential with $k = \alpha_j^{-1}, \ell = 1$ (i.e., we raise all group elements of the credential to the power α_j^{-1}; note that these numbers are known to the users), we can remove the number α from our considerations, so we will henceforth simply write

$$h_j = g_0 g_1^{x_{1,j}} \cdots g_n^{x_{n,j}}.$$

Let us write $\tilde{g}_i = g_i^f$. Then we can write $h_{3,j}$ as

$$h_{3,j} = h_j^{b_{1,j}} h_{2,j}^{b_{2,j}} = g_0^{b_{1,j}} \tilde{g}_0^{b_{2,j}} g_1^{b_{1,j} x_{1,j}} \tilde{g}_1^{b_{2,j} x_{1,j}} \cdots g_n^{b_{1,j} x_{n,j}} \tilde{g}_n^{b_{2,j} x_{n,j}}.$$

Setting $x_{0,j} = 1$ and writing $y_{i,j} = b_{1,j}x_{i,j}$ and $\tilde{y}_{i,j} = b_{2,j}x_{i,j}$, we get

$$h_{3,j} = g_0^{y_{0,j}} \tilde{g}_0^{\tilde{y}_{0,j}} g_1^{y_{1,j}} \tilde{g}_1^{\tilde{y}_{1,j}} \cdots g_n^{y_{n,j}} \tilde{g}_n^{\tilde{y}_{n,j}}, \tag{8.1}$$

where all numbers $y_{i,j}$ and $\tilde{y}_{i,j}$ are known to the user.

We know that $h_{4,j} = h_{3,j}^z$, i.e., the discrete log of $h_{4,j}$ with respect to $h_{3,j}$ is z. If we raise $h_{3,j}$ to some power and we simultaneously raise $h_{4,j}$ to the same power, then the resulting two elements will still have z as discrete log. The same holds if we multiply two elements $h_{3,j}$ and $h_{3,j'}$ together. In the remainder of this section we will take a number of powers and products of the elements $h_{3,j}$; whenever we write such a power or product, the same power or product for $h_{4,j}$ is implied.

Observe that when raising $h_{3,j}$ to the power $1/\tilde{y}_{n,1}$ we obtain a product of the generators g_i to certain exponents, where $\tilde{g}_n$ now has exponent 1. Thus two users 1 and 2 can work together to form the element $h_{3,1}^{1/\tilde{y}_{n,1}} / h_{3,2}^{1/\tilde{y}_{n,2}}$, which is of the form

$$\frac{h_{3,1}^{1/\tilde{y}_{n,1}}}{h_{3,2}^{1/\tilde{y}_{n,2}}} = g_0^{v_0} \tilde{g}_0^{\tilde{v}_0} g_1^{v_1} \tilde{g}_1^{\tilde{v}_1} \cdots g_n^{v_n}, \tag{8.2}$$

with $v_i = y_{i,1}/y_{n,1} - y_{i,2}/y_{n,2}$, and similar for $\tilde{v}_i$. Note that the right hand side no longer contains $\tilde{g}_n$. If two more users do the same and obtain a similar expression, then the four users can collectively remove g_n in exactly the same fashion, resulting in an expression as above containing only the elements $g_0, \tilde{g}_0, \ldots, g_{n-1}, \tilde{g}_{n-1}$.

Continuing in this fashion, 2^{2n+1} users can find an element in G_1 that is just g_0 raised to some power which is known and can easily be removed. If they apply all powers and products in parallel to the corresponding $h_{4,j}$, then they also obtain g_0^z. Similarly, they can obtain $\tilde{g}_0 = g_0^f$, and $\tilde{g}_0^z = g_0^{fz}$. In fact, they can do this for all elements $g_i, \tilde{g}_i$, resulting finally in expressions for g_i^f, g_i^z and g_i^{fz} for all i. Using these elements, anyone can calculate a valid credential over any set of attributes as explained below. The amount of users that need to work together to achieve this (2^{2n+1}) is exponential in n (the amount of attributes of the system) but *not* in the security parameter. Therefore, this can be done in polynomial time.

Remark 8.1. An alternative explanation for why this is possible is as follows. Suppose we are given m valid credentials, with $h_{3,j}$ of credential j given by (8.1). Notice that the operations we apply to the elements $h_{3,j}$ and $h_{4,j}$ above correspond exactly to taking linear combinations of the $h_{3,j}$ and $h_{4,j}$ (although linear combinations are usually written additively instead of multiplicatively). So if we consider the elements $g_i, \tilde{g}_i$ occurring in $h_{3,j}$ as unknowns, then we can interpret equation (8.1) as one equation in $2n+2$ unknowns. Thus if we have $2n+2$ credentials, then we obtain $2n+2$ equations in as many unknowns.

Using linear algebra over the field $\mathbb{Z}_q = \mathrm{GF}(q)$, then, we can solve this system of linear equations to the $g_i, \tilde{g}_i, g_i^{fz} = \tilde{g}_i^z$, as long as the $(2n+2) \times (2n+2)$ matrix of the

coefficients,

$$M := \begin{pmatrix} y_{0,1} & \cdots & y_{0,2n+2} \\ \tilde{y}_{0,1} & \cdots & \tilde{y}_{0,2n+2} \\ \vdots & \ddots & \vdots \\ y_{n,1} & \cdots & y_{n,2n+2} \\ \tilde{y}_{n,1} & \cdots & \tilde{y}_{n,2n+2} \end{pmatrix}$$

is invertible (i.e., its determinant $\det M$ is unequal to 0). Since the numbers $y_{i,j}, \tilde{y}_{i,j}$ are under our control in a chosen-message attack, this should be easy to achieve. If we write $m_{i,j}$ for the j-th entry of the i-th row of the inverse M^{-1} of M, we obtain

$$g_i = \prod_{j=1}^{2n+2} h_{3,j}^{m_{2i+1,j}}, \quad \tilde{g}_i = \prod_{j=1}^{2n+2} h_{3,j}^{m_{2i+2,j}}, \quad g_i^z = \prod_{j=1}^{2n+2} h_{4,j}^{m_{2i+1,j}}, \quad \tilde{g}_i^z = \prod_{j=1}^{2n+2} h_{4,j}^{m_{2i+2,j}}.$$

This also shows that the scheme is already completely forgeable (in the sense that new credentials with arbitrary attributes can be computed) with just $2n + 2$ collaborating users, instead of 2^{2n+1}.

8.3.2 Constructing a forged credential

Using the elements $g_i, g_i^f, g_i^z, g_i^{fz}$ constructed above, a new credential with attributes $x_1, \ldots, x_n$ may be constructed as follows. Choose $b_1, b_2 \in_R \mathbb{Z}_q$ randomly, and set

$$\begin{aligned} h &= g_0 g_1^{x_1} \cdots g_n^{x_n}, \\ h_2 &= g_0^f (g_1^f)^{x_1} \cdots (g_n^f)^{x_n}, \\ h_3 &= g_0^{b_1} (g_0^f)^{b_2} g_1^{b_1 x_1} (g_1^f)^{b_2 x_1} \cdots g_n^{b_1 x_n} (g_n^f)^{b_2 x_n}, \\ h_4 &= (g_0^z)^{b_1} (g_0^{fz})^{b_2} (g_1^z)^{b_1 x_1} (g_1^{fz})^{b_2 x_1} \cdots (g_n^z)^{b_1 x_n} (g_n^{fz})^{b_2 x_n}. \end{aligned}$$

Then

$$h_2 = h^f, \qquad h_3 = h^{b_1} h_2^{b_2}, \qquad h_4 = (h^{b_1} h_2^{b_2})^z$$

as required.

8.4 Analysis

8.4.1 The problem in the unforgeability argument

An argument for unforgeability is given in [HK14] in section 4, "Security Analysis". The argument is based on the appendix from [Ver01], in which it is argued that credentials

of the form

$$h,\ h_2 = h^f,\ h_4 = (h^{b_1} h_2^{b_2})^z \tag{8.3}$$

are unforgeable. Here, as above, f and z are the issuer's secret key, and the numbers b_1, b_2 are part of the credential (i.e., known to the user). However, the difference with Verheul's system is that there h is randomly chosen from G_1, and in particular, no participant of the system knows the discrete log of h with respect to any other element from G_1, or any DL-representation of h (i.e., an expression of h in terms of powers of $g_0, \ldots, g_n$, such as (8.4)). By contrast, in Hanzlik and Kluczniak's U-Prove scheme the user knows numbers $\alpha, x_1, \ldots, x_n$ such that

$$h = (g_0 g_1^{x_1} \cdots g_n^{x_n})^{\alpha}. \tag{8.4}$$

where the elements $g_0, \ldots, g_n$ are the same for all users. In this case, the argument from [Ver01] does not apply, so that no argument can be based on it.

In addition, we wish to point out that the argument from the appendix in [Ver01] was meant as a sketch, and in particular, there is the following subtlety. It is argued in the appendix that if an adversary $\mathcal{A}$ manages to forge credentials of the form (8.3), i.e.

$$(h, h_2, h_4, b_1, b_2) = \mathcal{A}\Big((h_j, h_{2,j}, h_{4,j}, b_{1,j}, b_{2,j})_{j=1,\ldots,m}\Big)$$

where the output (h, h_2, h_4, b_1, b_2) is valid (i.e., satisfying (8.3)), then either there must exist a j and numbers $k, \ell \in \mathbb{Z}_q$ such that

$$(h, h_2, h_4, b_1, b_2) = (h_j^k, h_{2,j}^k, h_{4,j}^{k\ell}, b_{1,j}\ell, b_{2,j}\ell)$$

or the adversary $\mathcal{A}$ can be used to solve discrete logarithms in G_1. However, the argument mentions certain "transformation factors" which are numbers like k, ℓ from $\mathbb{Z}_q$, and the algorithm sketched by [Ver01] that uses the adversary $\mathcal{A}$ to compute discrete logarithms would need to know these numbers in order to be able to work. However, it is not clear how to obtain these transformation factors from the adversary $\mathcal{A}$, or even if $\mathcal{A}$ is aware of them.

Notice, however, that a tuple (h, h_2, h_4, b_1, b_2) is very close to being an LRSW-instance (see Definition 9.1 on p. 155); indeed, if the number b_1 would always equal 1 then this would exactly be a LRSW-instance. For that reason, we believe that a (Type 1 version of) the Known Exponent Assumption (KEA, see Section 9.5 on p. 173) can be used to extract these numbers from the adversary, similar to how the difficulty of creating new LRSW-instances can be proven from the XKEA assumption (Theorem 9.21). This would result in a rigorous proof.

8.4.2 Why Theorem 6.6 is not applicable

In the notations of Theorem 6.6 on p. 118, we have the following.

- The space $\mathcal{P}$ of secret keys is the set of attributes $\mathbb{Z}_p^n \ni (x_1, \ldots, x_n)$.

- The space $\mathcal{K}$ of public keys consists of elements of the form $g_0 g_1^{x_1} \cdots g_n^{x_n}$ – that is, $\mathcal{K} = G_1$.
- The space $\mathcal{B}$ of blinding factors is $\mathbb{Z}_p^2 \ni (k, \ell)$.
- The space $\mathcal{S}$ is $G_1^2 \times \mathbb{Z}_p^2 \ni (h_2, h_4, b_1, b_2)$. This set has a natural group structure, namely the direct product: if $(A_1, A_2, r_1, r_2), (B_1, B_2, s_1, s_2) \in \mathcal{S}$ then

  $$(A_1, A_2, r_1, r_2) \cdot (B_1, B_2, s_1, s_2) = (A_1 B_1, A_2 B_2, r_1 + s_1, r_2 + s_2);$$

  i.e., for each element in the tuple we use the group structure of the containing group.
- The function $\mathsf{PubKey} \colon \mathcal{P} \times \mathcal{B} \to \mathcal{K}$ is $\mathsf{PubKey}((x_1, \ldots, x_n), (k, \ell)) = (g_0 g_1^{x_1} \cdots g_n^{x_n})^k$ (the number ℓ does not occur in the right hand side; it is used only in the function $\mathsf{Sig}_{f,z}$).
- The function $\mathsf{Sig}_{f,z} \colon \mathcal{P} \times \mathcal{B} \to \mathcal{S}$ is given by

  $$\mathsf{Sig}_{f,z}((x_1, \ldots, x_n), (k, \ell)) = (h^{fk}, (h^{kb_1\ell} h^{fkb_2\ell})^z, b_1\ell, b_2\ell)$$

  where $h = g_0 g_1^{x_1} \cdots g_n^{x_n}$.

Let $j = 1, 2$, and let

$$((x_{1,j}, \ldots, x_{n,1}), (h_j, h_{2,j}, h_{3,j}, h_{4,j}, b_{1,j}, b_{2,j})),$$
$$h_j = g_0 g_1^{x_{1,j}} \cdots g_n^{x_{n,j}}, \qquad h_{3,j} = h_j^{b_{1,j}} h_{2,j}^{b_{2,j}}$$

be two credentials. Translated to multiplicative notation, Theorem 6.6 studies in what situations there exist $(y_1, \ldots, y_n) \in \mathcal{P}$ and $(\alpha, \beta) \in \mathcal{B}$ such that the two equations

$$\begin{aligned} &\mathsf{PubKey}((y_1, \ldots, y_n), (\alpha, \beta)) \\ &\qquad = \mathsf{PubKey}((x_{1,1}, \ldots, x_{n,1}), (k_1, \ell_1))\mathsf{PubKey}((x_{1,2}, \ldots, x_{1,2}), (k_2, \ell_2)) \quad (8.5) \\ &\mathsf{Sig}_{SK}((y_1, \ldots, y_n), (\alpha, \beta)) \\ &\qquad = \mathsf{Sig}_{SK}((x_{1,1}, \ldots, x_{n,1}), (k_1, \ell_1))\mathsf{Sig}_{SK}((x_{1,2}, \ldots, x_{1,2}), (k_2, \ell_2)) \quad (8.6) \end{aligned}$$

hold simultaneously. The reason why the theorem from [HLR15] does not apply in this case is that if the upper equation holds then the lower equation never holds, as we now show.

First we examine equation (8.5). Using the group structure of G_1 on $\mathcal{K} = G_1$, the right hand side of (8.5) evaluates to $h_1^{k_1} h_2^{k_2}$ (and indeed it is not difficult to find $(y_1, \ldots, y_n), (\alpha, \beta)$ such that $\mathsf{PubKey}((y_1, \ldots, y_n), (\alpha, \beta)) = h_1^{k_1} h_2^{k_2}$). Now, the left hand side of (8.6) is a signature with $h_1^{k_1} h_2^{k_2}$ as its attribute commitment. The right hand side,

however, evaluates to (h_2, h_4, b_1, b_2) with

$$\begin{aligned} h_2 &= h_{2,1}^{k_1} h_{2,2}^{k_2} = \left(h_{1,1}^{k_1} h_{1,2}^{k_2}\right)^f, \\ h_4 &= \left(\left(h_1^{b_{1,1}} h_{2,1}^{b_{2,1}}\right)^{k_1\ell_1} \left(h_2^{b_{1,2}} h_{2,2}^{b_{2,2}}\right)^{k_2\ell_2}\right)^z, \\ b_1 &= \ell_1 b_{1,1} + \ell_2 b_{1,2}, \qquad b_2 = \ell_1 b_{2,1} + \ell_2 b_{2,2}. \end{aligned} \tag{8.7}$$

For this to be valid, h_4 would have to be

$$h_4 = \left(h^{b_1} h_2^{b_2}\right)^z = \left(\left(h_1^{k_1} h_2^{k_2}\right)^{\ell_1 b_{1,1} + \ell_2 b_{1,2}} \left(h_{2,1}^{k_1} h_{2,2}^{k_2}\right)^{\ell_1 b_{2,1} + \ell_2 b_{2,2}}\right)^z.$$

A polynomial-time algorithm can make the above expression for h_4 match with (8.7) only by matching the powers of the elements $h_1, h_{2,1}, h_2, h_{2,2}$. This gives

$$\begin{aligned} b_{1,1}k_1\ell_1 &= b_{1,1}k_1\ell_1 + b_{1,2}k_1\ell_2, & b_{1,2}k_2\ell_2 &= b_{1,2}k_2\ell_2 + b_{1,1}k_2\ell_1, \\ b_{2,1}k_1\ell_1 &= b_{2,1}k_1\ell_1 + b_{2,2}k_1\ell_2, & b_{2,2}k_2\ell_2 &= b_{2,2}k_2\ell_2 + b_{2,1}k_2\ell_1. \end{aligned}$$

which simplifies to

$$\begin{aligned} b_{1,2}k_1\ell_2 &= 0, & b_{1,1}k_2\ell_1 &= 0, \\ b_{2,2}k_1\ell_2 &= 0, & b_{2,1}k_2\ell_1 &= 0. \end{aligned}$$

We may assume that none of the numbers $b_{1,1}$, $b_{1,2}$, $b_{2,1}$, $b_{2,2}$ equal 0, as otherwise one or both of the signatures on the right hand side of (8.6) would not be valid. Thus it follows that at least one of the following must hold:

$$k_1 = k_2 = 0, \qquad k_1 = \ell_1 = 0, \qquad k_2 = \ell_2 = 0, \qquad \ell_1 = \ell_2 = 0.$$

By (8.7), the first and the last of these would result in $h_4 = 1$ which is not valid, and the middle two would completely remove one of the factors in the right hand side of (8.6), making the equation trivial.

8.4.3 The attack

Our attack depends on the following two facts.

- As mentioned in the introduction of this chapter, all credentials share the same base points $g_0, \ldots, g_n$.
- Since the prime order q of the group G_1 that contains the elements $g_0, \ldots, g_n$ and h, h_2, h_3, h_4 is known to all participants of the scheme (unlike in Idemix, where the order of the group is the issuer's private key), it is possible to invert exponents as in equation (8.2).

The combination of these two facts results in the ability to create signatures on DL-representations of successively less base elements, as in equation (8.2). In the unlinkable scheme that we introduce in Chapter 9, we avoid susceptibility to this attack by essentially letting each credentials have its own set of base points.

Chapter 9

An efficient self-blindable attribute-based credential scheme

Recently full-fledged implementations of unlinkable attribute-based credential schemes on smart cards have emerged. However, these need to compromise on the security level to achieve reasonable transaction speeds. In this chapter we present a new unlinkable attribute-based credential scheme. Defined on elliptic curves, the scheme involves bilinear pairings but only on the verifier's side, making it very efficient both in terms of speed and size on the user's side. We prove that the ShowCredential protocol of our scheme is a zero-knowledge proof of possession of a valid credential, from which unlinkability follows, and we provide two unforgeability proofs, one based on the LRSW assumption and the other on the Known Exponent Assumption (together with an extension of the discrete logarithm problem to Type 3 bilinear pairings). Finally, we briefly discuss the performance of a preliminary implementation.

An early version of the credential scheme from this chapter, together with an unforgeability proof based on the XKEA assumption was created jointly by Eric Verheul and myself, as an extension of Verheul's earlier article on self-blindable credentials [Ver01]. The additional unforgeability proof based on the whLRSW assumption is my own work, as well as the zero-knowledgeness and unlinkability proofs, and the implementation. I have also written the text below, with many helpful improvements and suggestions from Jaap-Henk Hoepman.

9.1 Introduction

As mentioned in the earlier chapters, the two most well-known attribute-based credential schemes are Idemix [CL01; IBM12] and U-Prove [Bra00; PZ13]. However, to date

there is no provably secure scheme that is sufficiently efficient to allow truly secure implementations on smart cards, while also providing unlinkability of transactions. For example, since Idemix is based on the strong RSA-problem, one would want the keysize to be at least 2048 bits and preferably even 4096 bits; the IRMA project[1] has implemented Idemix on smart cards using 1024 bits. On the other hand, U-Prove is more efficient but does not provide unlinkability; in addition, its security is not fully proven, and it has even been suggested that its unforgeability is unprovable under standard intractability assumptions [BL13b].

In this chapter, we provide a new provably secure, efficient and unlinkable attribute-based credential scheme, that is based on the concept of *self-blindability* [Ver01]: before showing the credential, it is randomly modified into a new one (containing the same attributes) that is still valid. This results in a showing protocol in which the verifier learns nothing at all about the credential besides the attributes that are disclosed (and the fact that the credential is valid). In fact, the showing protocol is a zero-knowledge proof of knowledge. The scheme does not rely on the random oracle model (although usage of this model can lead to a performance increase through the Fiat-Shamir heuristic [FS87]), and it uses elliptic curves and bilinear pairings, allowing the same security level as RSA-type groups at much smaller key sizes. Although computing a pairing is a much more expensive operation than performing exponentiations on an elliptic curve, all pairings occur on the verifier's side. In addition, the kinds of pairing that we use (Type 3; see Definition 5.11 on p. 91) involves two distinct groups of which one is more expensive to do computations on. However, the user only needs to perform computations on the cheaper of the two. As a consequence of these two facts the amount of work that the user has to perform is limited, and indeed our scheme is cheaper for the user than any comparable scheme that we know of.

Apart from the properties mentioned in Section 5.7, the credential scheme defined in this chapter will offer the following two kinds of unlinkability (see also Definition 5.28 on p. 108).

- *Multi-show unlinkability*: If a verifier participates in the ShowCredential protocol twice, in which the same credential was involved, it should be impossible for it to tell whether both executions originated from the same credential or from two different ones.
- *Issuer unlinkability*: If in a run of the ShowCredential protocol certain attributes were disclosed, then of all credentials that the issuer issued with those attributes, the issuer cannot tell which one was used.

We will achieve this by proving that our ShowCredential protocol is *black-box zero-knowledge* (see Definition 5.17 on p. 96), which essentially means that the verifier learns nothing at all besides the statement that the user proves. Since the verifier learns nothing that it can use to link transactions, both kinds of unlinkability follow from this (see Theorem 9.15).

The unforgeability of our credential scheme will be implied by the LRSW assumption [CL04; Lys+00; Lys99] introduced by Lysyanskaya, Rivest, Sahai, and Wolf, and

[1] https://www.irmacard.org

used in many subsequent works (for example, [CL04; WY05; Wac+11; CHP07; ACM05]). Actually, for our purposes a weaker (in particular, non-interactive and thus falsifiable [Nao03]) version of this assumption called the whLRSW assumption [WY05] will suffice. After having defined attribute-based credential schemes as well as unforgeability and unlinkability in the next section, we will discuss these assumptions in Section 9.2. In the same section we will introduce a signature scheme on the space of attributes, that will serve as the basis for our credential scheme. In Section 9.3 we turn to our credential scheme, defining issuing and showing protocols, and proving that these provide unlinkability and unforgeability for our scheme. This in turn implies the unforgeability of the signature scheme. In Section 9.5 we prove a general theorem saying that the whLRSW assumption is implied by (an extension of) the Known Exponent Assumption (KEA), resulting in a second unforgeability proof for our scheme. In Section 9.4 we will discuss the performance of our scheme, by counting the amount of exponentiations that the user has to perform and by showing average runtimes of an implementation of our scheme. First, we briefly review and compare a number of other attribute-based credential schemes, in terms of features, efficiency and speed, and security.

9.1.1 Related work

The Idemix credential scheme [CL01; IBM12] by Camenisch and Lysyanskaya is probably the most well-known unlinkable attribute-based credential scheme, relying on the difficulty of the strong RSA problem in the group of integers modulo an RSA modulus $n = pq$, of recommended size at least 2048 bits. Although this credential scheme has a lot of desirable properties (it is provably unlinkable and unforgeable), the large size of the modulus means that, when implementing the user on smart cards, it is difficult to get acceptable running times for the protocols. For example, in [VA13] the Idemix showing protocol has been implemented with 4 attributes and n around 1024 bits (while n should really be at least 2048 bits); there the running time for the ShowCredential protocol ranged from 1 to 1.3 seconds, depending on the amount of disclosed attributes.

Another well-known credential scheme is U-Prove [Bra00; PZ13] by Brands. Based on the difficulty of the discrete logarithm problem in a cyclic group, it can be implemented using elliptic curves, and additionally the showing protocol is much less complicated than that of Idemix, also resulting in more efficiency. However, in U-Prove two transactions executed with the same credential are always linkable, and the showing protocol is only honest-verifier zero-knowledge (i.e., there is no proof that dishonest verifiers cannot extract or learn information about the undisclosed attributes). Moreover, there is no unforgeability proof for U-Prove credentials, and it even seems that no such proof exists under standard intractability assumptions [BL13b].

We also mention the "Anonymous Credentials Light" construction from [BL13a], which can also be implemented on elliptic curves, but the credentials are not unlinkable; and [HM13], which runs in RSA groups like Idemix.

The credential scheme from [CL04], also by Camenisch and Lysyanskaya, is much closer to the scheme presented here: it is unlinkable, uses elliptic curves and (Type 1) pairings, and uses the LRSW assumption. However, when showing a credential the user has to compute an amount of pairings that is linear in the amount of disclosed attributes.

Finally, in Chapter 8 we showed an attack on the blindable U-Prove scheme from [HK14]: if sufficiently many users collide then they can create new credentials containing any set of attributes of their choice, without any involvement of the issuer.

9.2 Preliminaries

9.2.1 The LRSW assumptions

The unforgeability of the credential and signature schemes defined in this chapter will depend on the *whLRSW assumption* [WY05], which as we will show below, is implied by the LRSW assumption [Lys99; Lys+00] introduced by Lysyanskaya et al. The latter assumption has been proven to hold in the generic group model [Sho97], and has been used in a variety of schemes (for example, [CL04; WY05; Wac+11; CHP07; ACM05]). Although this assumption suffices to prove unforgeability of our scheme, it is stronger than we need. In particular, the LRSW assumption is an interactive assumption, in the sense that the adversary is given access to an oracle which it can use as it sees fit. We prefer to use the weaker whLRSW assumption, which is implied by the LRSW assumption but does not use such oracles. Consequentially, unlike the LRSW assumption itself, and like conventional hardness assumptions such as factoring and DDH, this assumption is falsifiable [Nao03]. We describe both assumptions below; then we prove that the LRSW assumption implies the whLRSW assumption. After this we will exclusively use the latter assumption.

Let $e\colon G_1 \times G_2 \to G_T$ be a Type 3 pairing (see Section 5.4 on p. 91), where the order p of the three groups is ℓ bits, and let $a, z \in_R \mathbb{Z}_p^*$. If $(\kappa, K, S, T) \in \mathbb{Z}_p^* \times G_1^3$ is such that $S = K^a$ and $T = K^{z+\kappa az}$, then we call (κ, K, S, T) an *LRSW-instance*.

Definition 9.1 (LRSW assumption). Let e be as above, and let $O_{a,z}$ be an oracle that, when it gets $\kappa_j \in \mathbb{Z}_p^*$ as input on the j-th query, chooses a random $K_j \in_R G_1$ and outputs the LRSW-instance $(\kappa_j, K_j, K_j^a, K_j^{z+\kappa_j az})$. The *LRSW problem* is, when given $(p, e, G_1, G_2, G_T, Q, Q^a, Q^z)$ where $Q \in_R G_2 \setminus \{1\}$, along with oracle access to $O_{a,z}$, to output a new LRSW-instance $(\kappa, K, K^a, K^{z+\kappa az})$ where κ has never been queried to $O_{a,z}$. The *LRSW assumption* is that no probabilistic polynomial-time algorithm can solve the LRSW problem with non-negligible probability in ℓ. That is, for every probabilistic polynomial-time algorithm $\mathcal{A}$ we have

$$\Pr\Big[a, z \in_R \mathbb{Z}_p^*;\ Q \in_R G_2 \setminus \{1\};\\ \sigma \leftarrow (p, e, G_1, G_2, G_T, Q, Q^a, Q^z);\quad (\kappa, K, S, T) \leftarrow \mathcal{A}^{O_{a,z}}(\sigma) :\\ K \in G_1 \ \wedge\ \kappa \in \mathbb{Z}_p^* \ \wedge\ \kappa \notin L \ \wedge\ S = K^a \ \wedge\ T = K^{z+\kappa az}\Big] < \mathrm{negl}(\ell),$$

where L is the list of oracle queries sent to $O_{a,z}$, and where the probability is over the choice of a, z, Q, and the randomness used by $\mathcal{A}$ and the oracle $O_{a,z}$.

Definition 9.2 (q-whLRSW assumption). Let e again be as above. Additionally, let $\{(\kappa_j, K_j, K_j^a, K_j^{z+\kappa_j az})\}_{j\in[1,q]}$ be a list of q LRSW-instances, where the κ_j and K_j are randomly distributed in $\mathbb{Z}_p^*$ and G_1, respectively. The *q-whLRSW problem* (for q-wholesale LRSW [WY05]) is, when given this list along with $(p, e, G_1, G_2, G_T, Q, Q^a, Q^z)$, to output a new LRSW-instance $(\kappa, K, K^a, K^{z+\kappa az})$ where $\kappa \notin \{\kappa_1, \ldots, \kappa_q\}$. The *$q$-whLRSW assumption* is that no probabilistic polynomial-time algorithm can solve the q-whLRSW problem with non-negligible probability in ℓ. That is, for every probabilistic polynomial-time algorithm $\mathcal{A}$ we have

$$\begin{aligned}\Pr\Big[&a, z, \kappa_1, \ldots, \kappa_q \in_R \mathbb{Z}_p^*;\ K_1, \ldots, K_q \in_R G_1 \setminus \{1\};\\ &Q \in_R G_2 \setminus \{1\};\ \sigma \leftarrow (p, e, G_1, G_2, G_T, Q, Q^a, Q^z);\\ &(\kappa, K, S, T) \leftarrow \mathcal{A}(\sigma, \{\kappa_j, K_j, K_j^a, K_j^{z+\kappa_j az}\}_{j\in[1,q]}) :\\ &K \in G_1 \ \wedge\ \kappa \in \mathbb{Z}_p^* \ \wedge\ \kappa \notin \{\kappa_1, \ldots, \kappa_q\} \ \wedge\ S = K^a \ \wedge\ T = K^{z+\kappa az}\Big] < \mathrm{negl}(\ell),\end{aligned} \tag{9.1}$$

where the probability is over the choice of $a, z, \kappa_1, \ldots, \kappa_q, K_1, \ldots, K_q, Q$, and the randomness used by $\mathcal{A}$.

Finally we define an unparameterized version of the assumption above by allowing q to be polynomial in ℓ, in the same way as the SDH problem is the unparameterized version of the q-SDH problem (see Definition 7.6 on p. 131). Intuitively, the reason that this unparameterized assumption is implied by the LRSW assumption is simple: if there is no adversary that can create LRSW-instances when it can (using the oracle) control the κ's of the LRSW-instances that it gets as input, then an adversary that can create them *without* having control over the κ's also cannot exist.

Definition 9.3. Let e, p and $\ell = |p|$ be as above. The *whLRSW assumption* states that for all positive polynomials $q\colon \mathbb{Z} \to \mathbb{N}$, the $q(\ell)$-whLRSW assumption holds.

Proposition 9.4. *The LRSW assumption implies the whLRSW assumption.*

Proof. Suppose that the whLRSW assumption does not hold, i.e., there is a polynomial q and a probabilistic polynomial-time algorithm $\mathcal{A}$ such that if $\mathcal{A}$ is given a list of $q(\ell)$ LRSW-instances, then it can produce a new valid LRSW-instance with non-negligible probability in ℓ. Now we create an algorithm $\mathcal{B}$ that violates the LRSW assumption. Algorithm $\mathcal{B}$ is given $\sigma = (p, e, G_1, G_2, G_T, Q, Q^a, Q^z)$ and oracle access to $Q_{a,z}$, and it operates as follows.

- It randomly chooses $q(\ell)$ values $\kappa_j \in_R \mathbb{Z}_p^*$;
- For each j, $\mathcal{B}$ calls $O_{a,z}(\kappa_j)$, obtaining a set of $q(\ell)$ valid LRSW-instances $L_j = (\kappa_j, K_j, K_j^a, K_j^{z+\kappa_j az})$;
- Algorithm $\mathcal{B}$ runs and returns the output of

$$\mathcal{A}(\sigma, L_1, \ldots, L_{q(\ell)}).$$

Then $\mathcal{B}$ is a probabilistic polynomial-time algorithm whose success probability is the same as that of $\mathcal{A}$, which is non-negligible by assumption. This contradicts the LRSW assumption. □

Thus if we prove that our scheme is safe under the whLRSW asssumption, then it is also safe under the LRSW assumption.

9.2.2 The discrete logarithm problem in bilinear group pairs

It is very common in cryptographic schemes to assume the difficulty of the discrete logarithm (DL) problem (Definition 5.6 on p. 89), and the difficulty of this problem is in fact implied by many other common hardness assumptions such as for example the computational and decisional Diffie-Hellman assumptions. In that sense, taking the DL assumption is one of the weakest assumption that one can take.

In our particular scheme, there are tuples (P, P^a, Q, Q^a) where $P \in G_1$, $Q \in G_2$. Thus we find ourselves interested in the following variation of the discrete logarithm problem:

$$\textit{Given } P, P^a, Q, Q^a \textit{ with } P \in G_1 \textit{ and } Q \in G_2\textit{, compute } a. \tag{9.2}$$

It is clear that in order for our scheme to be secure, this problem must be hard. This results in the following hardness assumption.

Definition 9.5 (BDL assumption). We say that the *bilinear discrete logarithm-assumption*, or the *BDL-assumption* for short, holds in a Type 3 bilinear pairing $e\colon G_1 \times G_2 \to G_T$, if there exists no probabilistic polynomial-time algorithm that can solve problem (9.2) above with non-negligible probability in $|p|$, where p is the order of the three groups.

The BDL assumption implies the DL assumption in G_1, G_2 and G_T, since if computing discrete logs in any of those groups would be easy then this problem would be easy too. However, the combination of the DL assumption in G_1, G_2 and G_T does not seem to imply the hardness of this problem.

We have not found this problem elsewhere in the literature. In a sense, however, this problem is not new at all. For example, it is implied by the LRSW and whLRSW assumptions; the q-SDH and the SDH assumptions (see [BB08], and Definition 7.6 on p. 131); as well as the co-CDH* assumption (which, when given P, P^a, Q, Q^a as above along with $R \in G_1$, asks for R^a), and a number of other pairing-specific variants of the Diffie-Hellman problem (however, it is not implied by the XKEA assumption; see Section 9.5.2). Indeed, such tuples P, P^a, Q, Q^a are present in many Type 3 schemes, so this problem needs to be hard in all of those schemes when a is private. This applies to, for example, the Boneh–Boyen signature scheme (see [BB08] and Chapter 7), the BLS signature scheme when adapted to Type 3 pairings (see [BLS04; SV07a] and Example 5.24 on p. 103), and the Boneh-Franklin identity-based encryption scheme in the Type 3 pairing setting [BF01; Gal05]. For those reasons, this seems a very natural assumption to take; indeed, it might be the weakest assumption that one can take in a Type 3 setting that implies the DL assumption in both G_1 and G_2 simultaneously.[2] We

[2] Alternatively, we could have used a Type 2 pairing instead of a Type 3 pairing, so that there would be an

speculate that it has not yet received any attention because it is implied by many other hardness assumptions, so that there is no need to explicitly consider it in schemes that use these stronger assumptions (even though it is true in such schemes). In that sense, the difficulty of this problem has been assumed all along.

For our purposes, the BDL assumption is important because of the following consequence. Recall that the DL assumption implies the difficulty of creating non-trivial DL-representations of 1 (see Proposition 5.7 on p. 89). The BDL assumption implies the following similar statement, that can be proven using an identical proof.

Proposition 9.6. *Let $e\colon G_1 \times G_2 \to G_T$ be a Type 3 pairing in which the BDL assumption holds, let p be the order of the three groups, and let P, Q be generators of G_1, G_2 respectively. Let the tuple $X_1, \ldots, X_m \in_R G_1$ and the tuple $Y_1, \ldots, Y_m \in G_2$ be such that $\log_P X_j = \log_Q Y_j$, i.e., $e(P, Y_j) = e(X_j, Q)$. Then no probabilistic polynomial-time algorithm can, on input $X_1, \ldots, X_m, Y_1, \ldots, Y_m$ generate a non-trivial DL-representation of $1 \in G_1$ with respect to $(X_1, \ldots, X_m)$.*

Note that any non-trivial DL-representation of 1 with respect to $(X_1, \ldots, X_m)$ in G_1 would also be one with respect to $(Y_1, \ldots, Y_m)$ in G_2. The only difference with Proposition 5.7 is that the adversary now also has $Y_1, \ldots, Y_m$ to work with.

9.2.3 A signature scheme on the space of attributes

In this section we introduce a signature scheme on the space of attributes. This signature scheme will be the basis for our credential scheme, in the following sense: the Issue protocol that we present in Section 9.3 will enable issuing such signatures over a set of attributes to users, while the ShowCredential protocol allows the user to prove that it has a signature over any subset of its signed attributes.

The Chaum-Pedersen signature scheme [CP93] serves as the basis for our signature scheme on the attribute space, and works as follows. Let $e\colon G_1 \times G_2 \to G_T$ be a bilinear group pair of Type 3, and let $Q \in G_2$ be a generator. The issuer's private and public keys are $a \in \mathbb{Z}_p$ and (e, A, Q) respectively, where $A = Q^a$. A signature on a message $K \in G_1$ is $S = K^a$, and it is verified by $e(S, Q) \stackrel{?}{=} e(K, A)$.

There are two immediate issues with this signature scheme. First, if (K, S) is a valid message-signature pair and $c \in \mathbb{Z}_p$, then (K^c, S^c) is also valid. Second, if (K_1, S_1) and (K_2, S_2) are both valid then (K_1K_2, S_1S_2) is also valid. That is, given valid message-signature pairs we can construct valid signatures on arbitrary powers and arbitrary products of the messages, without knowing the secret key. This signature scheme, then, is definitely not existentially unforgeable. On the other hand, this ability to modify valid signatures into new ones will allow us to create a showing protocol for credentials that is zero-knowledge in Section 9.3.

Definition 9.7 (Signature scheme on attribute space). The signature scheme is as follows.

efficiently computable isomorphism $\psi\colon G_2 \to G_1$. In that case the DL assumption in G_2 would have sufficed; in fact, it is easy to see that in this case the DL assumption in G_2 implies the BDL assumption. However, Type 3 pairings generally offer superior performance, and it has been suggested that most or all cryptographic schemes that use Type 2 pairings can be converted to Type 3 pairings [SV07b].

KeyGen$(1^\ell, n)$ The issuer generates a Type 3 pairing $e\colon G_1 \times G_2 \to G_T$, such that $|p| = \ell$ where p is the prime order of the three groups. Next it takes a generator $Q \in_R G_2$, and numbers $a, a_0, \ldots, a_n, z \in_R \mathbb{Z}_p^*$ and sets $A = Q^a, A_0 = Q^{a_0}, \ldots, A_n = Q^{a_n}$, and $Z = Q^z$. The public key is the tuple $PK = (p, e, Q, A, A_0, \ldots, A_n, Z)$ and the private key is the tuple $SK = (a, a_0, \ldots, a_n, z)$.

Sign$_{SK}(k_0, \ldots, k_n)$ The issuer chooses $\kappa \in_R \mathbb{Z}_p^*$ and $K \in_R G_1$, and sets $S = K^a, S_0 = K^{a_0}, \ldots, S_n = K^{a_n}$, and $T = (KS^\kappa \prod_{i=0}^n S_i^{k_i})^z$. The signature is $(\kappa, K, S, S_0, \ldots, S_n, T)$.

Verify$_{PK}((k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T))$ Setting $C = KS^\kappa \prod_{i=0}^n S_i^{k_i}$, the signature is verified by checking that $K, C \neq 1$, as well as

$$e(T, Q) \stackrel{?}{=} e(C, Z), \qquad e(S, Q) \stackrel{?}{=} e(K, A),$$
$$e(S_i, Q) \stackrel{?}{=} e(K, A_i) \qquad \text{for each } i = 0, \ldots, n.$$

The numbers $k_n \in \mathbb{Z}_p$ are the attributes. Although p may vary each time the KeyGen$(1^\ell, n)$ algorithm is invoked on a fixed security parameter ℓ, the attribute space $\mathbb{Z}_p$ will always contain $\{0, \ldots, 2^{\ell-1}\}$. In our credential scheme in section 9.3, the zeroth attribute k_0 will serve as the user's secret key, but at this point it does not yet have a special role.

Although the element $C = KS^\kappa \prod_{i=0}^n S_i^{k_i}$ is, strictly speaking, not part of the signature and therefore also not part of the credential (since it may be calculated from κ, the attributes $(k_0, \ldots, k_n)$ and the elements $(K, S, S_0, \ldots, S_n)$), we will often think of it as if it is. Finally, we call a message-signature pair, i.e., a tuple of the form $((k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T))$ where $(\kappa, K, S, S_0, \ldots, S_n, T)$ is a valid signature over $(k_0, \ldots, k_n)$, a *credential*.

Notice that if $(k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T)$ is a valid credential, then $(k_0, \ldots, k_n)$, $(\kappa, K^\alpha, S^\alpha, S_0^\alpha, \ldots, S_n^\alpha, T^\alpha)$ for any $\alpha \in \mathbb{Z}_p^*$ is another valid credential having the same attributes. That is, in the terminology of Verheul [Ver01] our credentials are *self-blindable*. At the same time, it is not at all clear how to change the credential into another one over different attributes, and indeed Theorem 9.8 essentially says that this is impossible. This self-blindability is what makes this signature scheme suitable for the purpose of creating an unlinkable ShowCredential protocol.

The number κ will play a critical role in the unforgeability proof of our signature and credential schemes (Theorem 9.13).[3]

Once the issuer has chosen an amount of attributes n and computed his public key, he can easily enlarge the maximum amount n of attributes that the scheme allows by generating additional secret keys $a_{n+1}, \ldots, a_{n'} \in_R \mathbb{Z}_p^*$, and adding the elements $A_{n+1} = Q^{a_{n+1}}, \ldots, A_{n'} = Q^{a_{n'}}$ to his public key. All credentials that have been computed beforehand will remain valid under this modified public key, and their value for the new attributes $k_{n+1}, \ldots, k_{n'}$ will be 0.[4]

[3] We could have eased the notation somewhat by denoting κ as an extra attribute k_{n+1}, but because it plays a rather different role than the other attributes (it is part of the signature), we believe this would create more confusion than ease.

[4] This means that the value 0 that the new attributes will have must not have any meaning; i.e., it should be clear to all participants in the scheme that this value indicates that the attribute was not issued to the credential

Table 9.1. Notation comparison between our signature scheme and the Camenisch-Lysyanskaya signature scheme [CL02].

Object	Ours	CL	Remarks
signature	T	A	
attributes	k_i	m_i	
randomizer	κ	v	
prime		e	specific to CL-scheme
randomizer base point	S	S	differs per credential in our scheme
attribute base points	S_i	R_i	differs per credential in our scheme
base point	K	Z	differs per credential in our scheme
public key elements	Q, A, A_i, Z		specific to our scheme

Theorem 9.8. *Our credentials are existentially unforgeable under adaptively chosen message attacks (in the sense of Definition 5.23 on p. 102), under the whLRSW assumption.*

We will prove this after we have proven unforgeability of our credential scheme (Theorem 9.13). Unforgeability also follows from the XKEA and BDL assumptions, as they together imply the whLRSW assumption (see Theorem 9.21).

Remark 9.9. Our signature scheme has a number of similarities with the Camenisch-Lysyanskaya signature scheme [CL02] on which Idemix is built. Most of the ingredients of our scheme have an analogue in Idemix that more or less serves the same purpose. Although the comparison is not always perfect and thus should not be taken too literally, we include a comparison of the various elements of both schemes in Table 9.1. Without presenting all details, in the Camenisch-Lysyanskaya signature scheme, the attributes are denoted with m_i and a signature on a set of attributes is a triple (A, e, v), which is such that $A^e = Z/(S^v R_0^{m_0} \cdots R_n^{m_n})$, or equivalently $A^{-1} = (Z^{-1} S^v R_0^{m_0} \cdots R_n^{m_n})^{1/e}$. Here e is a prime and $Z, S, R_0, \ldots, R_n$ are part of the issuer public key. Recall that in our scheme, a signature is a tuple $(\kappa, K, S, S_0, \ldots, S_n, T)$ such that $T = (K S^\kappa S_0^{k_0} \cdots S_n^{k_n})^z$.

9.3 The credential scheme

In this section we present our credential scheme. The strategy is as follows: having defined an unforgeable signature scheme on the set of attributes $\mathbb{Z}_p^n$ (Definition 9.7), we provide an issuing protocol, in which the issuer grants a credential to a user, and a showing protocol, which allows a user to give a zero-knowledge proof to a verifier that he possesses a credential, revealing some of the attributes contained in the credential while keeping the others secret. The Issue protocol is shown in Figure 9.1, and the ShowCredential protocol is shown in Figure 9.2. Here and in the remainder of the chapter, we will write $\mathcal{D} \subset \{1, \ldots, n\}$ for the index set of the disclosed attributes, and

$$\mathcal{C} = \{1, \ldots, n\} \setminus \mathcal{D}$$

owner.

Common information: Attributes $k_1, \ldots, k_n$, issuer's public key $PK = (p, e, Q, A, A_0, \ldots, A_n, Z)$

User knows secret key k_0		**Issuer** knows $SK = (a, a_0, \ldots, a_n, z)$
		choose $\bar{K} \in_R G_1$
	$\longleftarrow$	send $\bar{S} = \bar{K}^a, \bar{S}_0 = \bar{K}^{a_0}$
choose $\alpha, \kappa' \in_R \mathbb{Z}_p^*$		
set $S = \bar{S}^\alpha, S_0 = \bar{S}_0^\alpha$		
send $S, S_0, R = S^{\kappa'} S_0^{k_0}$	$\longrightarrow$	
$\mathrm{PK}\{(\kappa', k_0) \colon R = S^{\kappa'} S_0^{k_0}\}$	$\longleftrightarrow$	
		set $K = S^{1/a}$
		verify $S \neq \bar{S}, K = S_0^{1/a_0}$
		choose $\kappa'' \in_R \mathbb{Z}_p$
		set $S_i = K^{a_i}\ \forall i \in [1, n]$
		set $T = \left(KS^{\kappa''} R \prod_{i=1}^n S_i^{k_i}\right)^z$
	$\longleftarrow$	send $\kappa'', K, S_1, \ldots, S_n, T$
set $\kappa = \kappa' + \kappa''$		
$\mathsf{Verify}_{PK}((k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T))$		
return $(k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T)$		

Figure 9.1. The Issue protocol. In the protocol, the issuers sends two elements $\bar{S}, \bar{S}_0$ (having the appropriate relative discrete log) to the user, who blinds them using a random number, and sends the blinded versions to the issuer. With respect to these blinded elements, the user then proves that it knows its secret key k_0 and its contribution κ' to the number κ. If the verifier is convinced, it chooses its own contribution κ'' to κ, and it computes the remaining elements $K, S_1, \ldots, S_n, T$ such that $(\kappa' + \kappa'', K, S, S_0, \ldots, S_n, T)$ is a valid signature over the attributes. These elements are sent to the user who finally constructs the credential.

for the index set of the undisclosed attributes. We do not consider the index 0 of the secret key k_0 to be an element of $\mathcal{C}$, as the secret key is always kept secret.

The Issue protocol is such that both parties contribute to κ and K with neither party being able to choose the outcome in advance (unlike the signing algorithm of the signature scheme from the previous section, where the signer chooses κ and K on its own). This ensures that these elements are randomly distributed even if one of the parties is dishonest. Additionally, the issuer is prevented from learning the values of κ and the secret key k_0.

As noted earlier, we assume that the user and issuer have agreed on the attributes $k_1, \ldots, k_n$ to be contained in the credential before executing this protocol. Similarly, we assume that the user sends the dislosure set $\mathcal{D}$ and disclosed attributes $(k_i)_{i \in \mathcal{D}}$ to the verifier prior to executing the ShowCredential protocol.

Remark 9.10. Although we have not explicitly denoted so in the protocols, it is important (in both protocols) that whenever one party receives a group element (say, $X = (x, y)$

Common information: Issuer's public key $PK = (p, e, Q, A, A_0, \ldots, A_n, Z)$; disclosure set $\mathcal{D}$, undisclosed set $\mathcal{C} = \{1, \ldots, n\} \setminus \mathcal{D}$; disclosed attributes $(k_i)_{i\in\mathcal{D}}$

	User		**Verifier**
knows $K, S, S_0, \ldots, S_n, \kappa, (k_i)_{i\in\mathcal{C}}, C, T$			
	choose $\alpha, \beta \in_R \mathbb{Z}_p^*$		
	set $\bar{K} = K^\alpha$, $\bar{S} = S^\alpha$, $\bar{S}_i = S_i^\alpha \ \forall i \in [0, n]$		
	set $\tilde{C} = C^{-\alpha/\beta}$, $\tilde{T} = T^{-\alpha/\beta}$		
	send $\bar{K}, \bar{S}, (\bar{S}_i)_{i=0,\ldots,n}, \tilde{C}, \tilde{T}$	$\longrightarrow$	
	set $D = \bar{K}^{-1} \prod_{i\in\mathcal{D}} \bar{S}_i^{-k_i}$		set $D = \bar{K}^{-1} \prod_{i\in\mathcal{D}} \bar{S}_i^{-k_i}$
$\mathrm{PK}\{(\beta, \kappa, k_0, k_i)_{i\in\mathcal{C}} : D = \tilde{C}^\beta \bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i\in\mathcal{C}} \bar{S}_i^{k_i}\}$		$\longleftrightarrow$	
			verify $e(\bar{K}, A) \stackrel{?}{=} e(\bar{S}, Q)$
			and $e(\bar{K}, A_i) \stackrel{?}{=} e(\bar{S}_i, Q)_{\forall i\in[0,n]}$
			and $e(\tilde{C}, Z) \stackrel{?}{=} e(\tilde{T}, Q)$

Figure 9.2. The ShowCredential protocol. We assume that the user has the element $C = KS^\kappa S_0^{k_0} \cdots S_n^{k_n}$ stored so that it does not need to compute it every time the protocol is run (see Section 9.4 for more such optimizations). In the protocol, he user first blinds K, S and each S_i with a random number, and C and T with a different random number, resulting in new elements $\bar{K}, \bar{S}, \bar{S}_i$ and $\tilde{C}, \tilde{T}$. These are sent to the verifier. Then, the user proves that he knows the hidden attributes and the number κ, as well as a number β which is such that $\tilde{C}^\beta$ is of the required form $\tilde{C}^\beta = \bar{K}\bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i=1}^n \bar{S}_i^{k_i}$. If the proof of knowledge is valid and the elements $\bar{K}$, $\bar{S}$ and $\bar{S}_i$ on the one hand and $\tilde{C}$, $\tilde{T}$ on the other hand have the appropriate relative discrete logarithms (which the verifier checks by calculating a number of pairings), then the verifier accepts.

from G_1 or G_2) from the other party, that it checks that X is indeed a valid element from G_1 or G_2. That is, it should be verified that the coordinates (x, y) of X satisfy the equation that defines the elliptic curve group. For example, if the curve is defined using a Weierstrass equation with coefficients a, b, then

$$y^2 = x^3 + ax + b$$

must hold, in order to prevent invalid curve attacks such as for example those from [Ant+02]. For the same reason, if the group G_1 or G_2 is a proper subgroup of some elliptic curve group, then it must also be checked that the element is indeed a member of this subgroup.

Mathematically, we can formalize what the ShowCredential protocol should do as follows. Let $\mathcal{K}_{p,n}$ be the set of all public keys having n attributes and where p is the order of the groups G_1, G_2 and G_T. The common knowledge of the user and verifier when running the ShowCredential protocol consists of elements of the following formal

language:

$$L = \{(PK, \mathcal{D}, (k_i)_{i\in\mathcal{D}}) \mid p, n > 0;\ p \text{ prime};\ PK \in \mathcal{K}_{p,n};\ \mathcal{D} \subset \{1,\dots,n\};\ k_i \in \mathbb{Z}_p\ \forall i \in \mathcal{D}\}. \quad (9.3)$$

In addition, let the relation R be such that $R(x,w) = 1$ only if $x = (PK, \mathcal{D}, (k_i)_{i\in\mathcal{D}}) \in L$, and $w = ((k'_0,\dots,k'_n), s)$ is a valid credential with respect to PK, with $k'_i = k_i$ for $i \in \mathcal{D}$ (i.e., the disclosed attributes $(k_i)_{i\in\mathcal{D}}$ are contained in the credential w.) Thus the equation $R(x,w) = 1$ holds only if w is a valid credential having attributes $(k_i)_{i\in\mathcal{D}}$.

In Theorems 9.11, 9.12 and 9.14 we will prove that our ShowCredential protocol is a black-box zero-knowledge proof of knowledge (see Definition 5.17) for this relation. From this fact unforgeability and unlinkability will follow.

Theorem 9.11. *The showing protocol is complete with respect to the language L: if a user has a valid credential then it can make the verifier accept.*

Proof. If the user follows the ShowCredential protocol, then $e(\bar{K}, A) = e(K^\alpha, Q^a) = e(K^{\alpha a}, Q) = e(S^\alpha, Q) = e(\bar{S}, Q)$, so the first verification that the verifier does will pass. An almost identical calculation shows that the second and third verifications pass as well. As to the proof of knowledge, setting $\bar{C} = C^\alpha$ we have

$$\tilde{C}^\beta \bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i\in\mathcal{C}} \bar{S}_i^{k_i} = \bar{C}^{-1} \bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i\in\mathcal{C}} \bar{S}_i^{k_i} = \bar{K}^{-1} \prod_{i\in\mathcal{D}} \bar{S}_i^{-k_i} = D, \quad (9.4)$$

so the user can perform this proof without problem. □

9.3.1 Unforgeability

Lemma 9.12. *With respect to the language L defined in (9.3), the* ShowCredential *protocol is black-box extractable, in the sense of Definition 5.17 on p. 96.*

Proof. By Definition 5.17, we must show the existence of an extractor χ satisfying the following: for all $x \in L$, if a probabilistic polynomial-time algorithm $\mathcal{P}^*$, acting as the user, can successfully run the ShowCredential protocol with a verifier with probability $\epsilon(|x|)$, then there is an algorithm χ that, when given black-box access to $\mathcal{P}^*$, extracts with probability $\epsilon(|x|) - \nu(|x|)$ a valid credential from $\mathcal{P}^*$, where $\nu(|x|)$ is some negligible function.

As part of our ShowCredential protocol, the user performs a zero-knowledge proof of knowledge of the numbers β, κ, k_0 and $(k_i)_{i\in\mathcal{C}}$ that are such that $D = \tilde{C}^\beta \bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i\in\mathcal{C}} \bar{S}_i^{k_i}$. By Definition 5.17, then, there exists an extractor χ_D that extracts these numbers from this zero-knowledge proof that fails with negligible probability $\nu_D(|x|)$. Our extractor χ uses this extractor χ_D, as follows:

- first it stores the group elements $\bar{K}, \bar{S}, (\bar{S}_i)_{i=0,\dots,n}, \tilde{T}$ that the user sends to it;

- it uses the extractor χ_D to obtain the numbers β, κ, k_0 and $(k_i)_{i\in\mathcal{C}}$ from the proof of knowledge;
- it sets $\bar{T} = \tilde{T}^{-\beta}$; note that if T was valid then we have $\bar{T} = T^{\alpha} = C^{\alpha z} = (\bar{K}\bar{S}^{\kappa}\bar{S}_0^{k_0}\prod_{i=0}^{n}\bar{S}_i^{k_i})^z$,
- it returns $(k_0,\ldots,k_n),(\kappa,\bar{K},\bar{S},\bar{S}_0,\ldots,\bar{S}_n,\bar{T})$.

The only action that χ performs that normal verifiers cannot perform is the extraction of the numbers $\beta,\kappa,k_0,(k_i)_{i\in\mathcal{C}}$ from the proof of knowledge. Therefore, if the user $\mathcal{P}^*$ convinces normal verifiers with probability $\epsilon(|x|)$, then χ will succeed with probability $\epsilon(|x|)-\nu_D(|x|)$. Since ν_D is negligible by Definition 5.17, this proves the claim. □

In the proof of the unforgeability theorem, we will need a tuple $(\hat{K},\hat{S},\hat{S}_0,\ldots,\hat{S}_n)\in G_1^{n+3}$ such that $\hat{S}=\hat{K}^a$ and $\hat{S}_i=\hat{K}^{a_i}$ for all i. For that reason we will henceforth assume that such a tuple is included in the issuer's public key (we will need this tuple also when proving unlinkability of our scheme in Theorem 9.14).[5] Notice that the presence of this tuple does not endow the adversary with any extra capabilities, as it could already obtain such a tuple in an Issue query to the challenger.

Theorem 9.13. *Our credential scheme is unforgeable under the whLRSW assumption, in the sense of Definition 5.27 on p. 107.*

Proof. Suppose that there exists an adversary $\mathcal{A}$ that can break unforgeability of our scheme, using q_I Issue queries and q_S ShowCredential queries. Then we build an algorithm $\mathcal{B}$ that contradicts the q-whLRSW assumption with $q=q_I+q_S$. As q must be polynomial in ℓ (otherwise adversary $\mathcal{A}$ could not be polynomial-time), this in turn contradicts the whLRSW assumption. Thus, adversary $\mathcal{A}$ plays the credential unforgeability game with algorithm $\mathcal{B}$, who acts as the challenger of $\mathcal{A}$ on the one hand and will use what it learns from this game to violate the q-whLRSW assumption on the other hand.

The first thing to notice is that if we take a setup of $n=0$ attributes, then credentials in this setup are LRSW instances. The public key of such a setup would be $Q,\bar{A}=Q^{\bar{a}},Z=Q^z$ (where the reason for the bar on top of the second group element will become clear shortly), and a valid credential would be $\bar{\kappa},K,\bar{S}=K^{\bar{a}},T=(K\bar{S}^{\bar{\kappa}})^z=K^{z+\bar{\kappa}\bar{a}z}$, which is indeed an LRSW instance. Suppose that algorithm $\mathcal{B}$ is given a list of $q=q_I+q_S$ such LRSW-instances

$$\bar{\kappa}_j,K_j,\bar{S}_j,T_j=(K_j\bar{S}_j^{\bar{\kappa}_j})^z=K_j^{z+\bar{\kappa}_j\bar{a}z},$$

along with $Q,\bar{A},Z$. Algorithm $\mathcal{B}$, acting as the challenger, engages in the unforgeability game with $\mathcal{A}$, acting as follows.

Setup Challenger $\mathcal{B}$ lets $\mathcal{A}$ decide on the number of attributes n. Next it generates $a,a_0,\ldots,a_n\in_R\mathbb{Z}_p^*$ and then sets $A=\bar{A}^a$ and $A_i=\bar{A}^{a_i}$ for $i=0,\ldots,n$. Then it

[5] Note that credential owners already have such a tuple; verifiers can obtain one simply by executing the ShowCredential protocol with a user that has a valid credential; and issuers can of course create such tuples by themselves. Therefore in practice, each party participating in the scheme will probably already have such a tuple, so that including it in the public key may not be necessary in implementations.

sends the public key

$$Q, A, A_0, \ldots, A_n, Z$$

to $\mathcal{A}$.

Queries Challenger $\mathcal{B}$ answers queries from $\mathcal{A}$ as follows. We use the index j here for both kinds of queries; i.e., if either query is performed then j is incremented by 1.
Issue$(k_{1,j}, \ldots, k_{n,j})$: Challenger $\mathcal{B}$ engages in the issuing protocol with $\mathcal{A}$ on the specified attributes. Using the LRSW-instance $\bar{\kappa}_j, K_j, \bar{S}_j, T_j$, note that the challenger can compute elements $S_{i,j}$ which are valid with respect to its public key by

$$S_j = \bar{S}_j^a \qquad \text{and} \qquad S_{i,j} = \bar{S}_j^{a_i}$$

for all $j = 0, \ldots, m$. Using these elements it performs the issuing protocol normally, but when the adversary proves knowledge of its private key $k_{0,j}$ and the number κ', $\mathcal{B}$ extracts these numbers from the proof of knowledge using the extractor guaranteed to exist by Definition 5.17. Next, it solves the equation

$$\kappa_j a + \sum_{i=0}^{n} k_{i,j} a_i = \bar{\kappa}_j \tag{9.5}$$

to κ_j (i.e., it sets $\kappa_j = (\bar{\kappa}_j - \sum_{i=0}^{n} k_{i,j} a_i)/a$), and uses the number $\kappa_j - \kappa'$ as the value for κ'' in the remainder of the protocol.
Notice that we then have

$$K_j \bar{S}_j^{\bar{\kappa}_j} = K_j \bar{S}_j^{\kappa_j a + \sum_{i=0}^{n} k_{i,j} a_i} = K_j S_j^{\kappa_j} \prod_{i=0}^{n} S_{i,j}^{k_{i,j}},$$

and

$$e(S_{i,j}, Q) = e(\bar{S}_j^{a_i}, Q) = e(K_j^{\bar{a} a_i}, Q) = e(K_j, \bar{A}^{a_i}) = e(K_j, A_i),$$

and similarly $e(S_j, Q) = e(K_j, A)$. This means that $(k_{0,j}, \ldots, k_{n,j}), (\kappa_j, K_j, S_j, S_{0,j}, \ldots, S_{n,j}, T_j)$ is a valid credential.
ShowCredential$(\mathcal{D}, k_{1,j}, \ldots, k_{n,j})$: The challenger $\mathcal{B}$ embeds $k_{1,j}, \ldots, k_{n,j}$ in one of its LRSW-instances as it does in Issue queries, randomly choosing a κ_j and $k_{0,j}$ itself. Next it performs the ShowCredential protocol with the adversary over $(k_{i,j})_{i \in \mathcal{D}}$ as requested.

Output When $\mathcal{A}$ and $\mathcal{B}$ engage in the ShowCredential protocol, $\mathcal{B}$ uses the extractor guaranteed to exist by Lemma 9.12 above, obtaining a credential

$$(k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T). \tag{9.6}$$

Then $\mathcal{B}$ calculates $\bar{\kappa} = \kappa a + \sum_{i=0}^{n} k_i a_i$ and outputs $\bar{\kappa}, K, \bar{S} = S^{1/a}, T$.

If the credential (9.6) that $\mathcal{B}$ extracts from $\mathcal{A}$ is valid, then $S = K^{\bar{a} a} = \bar{S}^a$ and $S_i = K^{\bar{a} a_i} =$

$\bar{S}^{a_i}$. Also, setting $\bar{\kappa} = \kappa a + \sum_{i=0}^{n} k_i a_i$, T equals

$$T = \left(K S^{\kappa} \prod_{i=0}^{n} S_i^{k_i}\right)^z = \left(K \bar{S}^{\kappa a + \sum_{i=0}^{n} k_i a_i}\right)^z = \left(K \bar{S}^{\bar{\kappa}}\right)^z = K^{z + \bar{\kappa} \bar{a} z}.$$

This implies that the tuple $\bar{\kappa}$, K, $\bar{S}$, T is a valid LRSW-instance. In order to derive a contradiction with the q-whLRSW assumption, it remains to show that with non-negligible probability $\bar{\kappa} \neq \bar{\kappa}_j$ for all j.

Suppose that there is a non-negligible chance that adversary $\mathcal{A}$ wins such that $\bar{\kappa} = \bar{\kappa}_j$ for some j. Then we have for this j

$$S^{\kappa} \prod_{i=0}^{n} S_i^{k_i} = \bar{S}^{\bar{\kappa}} = \bar{S}^{\bar{\kappa}_j} = S^{\kappa_j} \prod_{i=0}^{n} S_j^{k_{i,j}},$$

or equivalently,

$$1 = S^{\kappa - \kappa_j} \prod_{i=0}^{n} S_i^{k_i - k_{i,j}}. \tag{9.7}$$

Now there are two possibilities: either j corresponds to an Issue query or to a ShowCredential query. If j was an Issue query, then it follows from the fact that $\mathcal{A}$ won that the DL-representation (9.7) of 1 above is nontrivial, otherwise the output of $\mathcal{A}$ would not have been a new credential. On the other hand, if j was a ShowCredential query, then the value of κ and k_0 that challenger $\mathcal{B}$ chose and used are information-theoretically hidden from $\mathcal{A}$. This means that the probability that $\mathcal{A}$ chose the values of κ, k_i exactly such that $\kappa = \kappa_j$ and $k_i = k_{i,j}$ is negligible, so that the DL-representation above will still be nontrivial with overwhelming probability. In both cases the DL-representation is thus nontrivial with overwhelming probability. Now, since the elements $\hat{S}, \hat{S}_0, \ldots, \hat{S}_n$ have the same relative discrete logarithms as the elements $S, S_0, \ldots, S_n$, this results in the following non-trivial DL-representation:

$$1 = \hat{S}^{\kappa - \kappa_j} \prod_{i=0}^{n} \hat{S}_i^{k_i - k_{i,j}}. \tag{9.8}$$

Notice that it is easy to check if this holds for some j even without knowledge of the secret key $a, a_0, \ldots, a_n, z$. We can therefore exploit this ability of the adversary without knowledge of these numbers to contradict Proposition 9.6 as follows. Let $P, X_0, \ldots, X_{n+1} \in G_1$, $Q, Y_0, \ldots, Y_{n+1} \in G_2$ be given, with $e(P, Y_i) = e(X_i, Q)$. We construct a non-trivial DL-representation of 1 with respect to $X_1, \ldots, X_n$ as follows.

- Set $\hat{K} = P, \hat{S} = X_0, \hat{S}_i = X_{i+1}$, and $A = Y_0, A_i = Y_{i+1}$. Take $z \in_R \mathbb{Z}_p^*$ and set $Z = Q^z$.
- Play the unforgeability game with the adversary with respect to the public key $(p, e, Q, A, A_0, \ldots, A_n, Z)$ constructed above. In each query, generate a new valid tuple $(K, S, S_0, \ldots, S_n)$ by taking a random number $r \in \mathbb{Z}_p^*$ and setting $K = \hat{K}^r$,

$S = \hat{S}^r$, and $S_i = \hat{S}_i^r$ for $i = 0, \ldots, n$. At the end of the game, return the resulting DL-representation (9.8) of 1.

As this would contradict Proposition 9.6, we conclude that we must have $\bar{\kappa} \neq \bar{\kappa}_j$ with overwhelming probability. But then the output $(\bar{\kappa}, K, \bar{S}, T)$ of algorithm $\mathcal{B}$ would be a new LRSW-instance, contradicting the q-whLRSW assumption. □

The unforgeability of the credential scheme implies that of the underlying signature scheme, as follows.

Proof of Theorem 9.8. Suppose that there exists an adversary $\mathcal{A}$ that can forge the signature scheme from Section 9.2.3. Then we create a forger $\mathcal{B}$ for our credential scheme as follows. Forger $\mathcal{B}$ is the challenger of adversary $\mathcal{A}$ in the signature scheme unforgeability game, and the adversary in the credential scheme unforgeability game. It operates as follows.

Setup Forger $\mathcal{B}$ receives a public key from its challenger and forwards it to the adversary $\mathcal{A}$.

Queries Whenever adversary $\mathcal{A}$ requests a signature on a set of attributes $(k_0, \ldots, k_n)$, $\mathcal{B}$ performs an Issue query on these attributes with its challenger. It sends the resulting signature to $\mathcal{A}$.

Output If $\mathcal{A}$ outputs a valid new credential then $\mathcal{B}$ uses this in the ShowCredential protocol with its challenger.

Then the success probability of $\mathcal{B}$ will be the same as that of $\mathcal{A}$. □

9.3.2 Anonymity

In order to prove that our ShowCredential protocol is zero-knowledge with respect to the language L from equation (9.3), we must show the existence of a simulator whose behavior is from the perspective of a verifier indistinguishable from an honest user. In order to achieve this, the simulator will need a pair $(\hat{C}, \hat{T}) \in G_1^2$ such that $\hat{T} = \hat{C}^z$ (as well as the elements $\hat{K}, \hat{S}, \hat{S}_0, \ldots, \hat{S}_n$ that we added to the public key earlier). For that reason, we will henceforth assume that such a pair is included with the public key PK of the credential scheme. Note that one can view these elements $\hat{K}, \hat{S}, \hat{S}_0, \ldots, \hat{S}_n, \hat{C}, \hat{T}$ as an extra credential of which the numbers $(\kappa, k_0, \ldots, k_n)$ are not known. Therefore the credential scheme remains unforgeable (in the sense that Theorem 9.13 still holds).

Theorem 9.14. *The* ShowCredential *protocol is a black-box zero-knowledge proof of knowledge with respect to the language L.*

Proof. The ShowCredential protocol is complete (Theorem 9.11), and extractable (Theorem 9.12), so by Definition 5.17 on p. 96 it remains to show here that there exists a simulator whose behavior is indistinguishable from an honest user. This simulator $\mathcal{S}$ is given the issuer's public key, a disclosure set $\mathcal{D}$, and a list of attributes k_i for $i \in \mathcal{D}$. We have it proceed as follows.

- It chooses random $\alpha, \beta \in_R \mathbb{Z}_p^*$;
- It sets $\bar{K} = \hat{K}^\alpha$, $\bar{S} = \hat{S}^\alpha$, $\bar{S}_i = \hat{S}_i^\alpha$ for $i = 0, \ldots, n$, and $\tilde{C} = \hat{C}^\beta$, $\tilde{T} = \hat{T}^\beta$;
- It sends these values to the verifier, and then uses the simulator from the proof of knowledge of the numbers $\beta, \kappa, k_0, (k_i)_{i \in \mathcal{C}}$.

It remains to show that this behavior is indistinguishable from that of honest users, to any verifier that is given any auxiliary information. First notice that for honest users and the simulator alike, the elements $\bar{K}$ and $\tilde{C}$ are always randomly distributed in G_1. Also, again for both honest users and the simulator, the elements $\bar{S}, \bar{S}_i$ and $\tilde{T}$ are determined by $\bar{K}$ and $\tilde{C}$ respectively.

Notice that for any $\beta \in \mathbb{Z}_p^*$ and any set $\mathcal{C}$ of undisclosed attributes, there exist numbers $\kappa, k_0, (k_i)_{i \in \mathcal{C}}$ such that equation (9.4) holds. Thus the only difference between honest users and the simulator is that an honest user knows these numbers and uses them to honestly prove knowledge of them, while the simulator simulates this proof. However, by the black-box zero-knowledge properties of the proof of knowledge over these numbers, this cannot be detected by the verifier. Thus the verifier can behave no different than it would have done if it had interacted with an honest user $\mathcal{P}$. □

Theorem 9.15. *Let* (KeyGen, Issue, ShowCredential) *be an attribute-based credential scheme whose* ShowCredential *protocol is black-box zero-knowledge. Then the scheme is unlinkable in the sense of Definition 5.28.*

Proof. Let the auxiliary input to the verifier be whatever it learns in the Queries phase of the unlinkability game. In the Challenge phase, instead of performing the showing protocol normally using credential j_b, the challenger uses the simulator $\mathcal{S}$ whose existence is guaranteed by the black-box zero-knowledge property of the ShowCredential protocol. It is clear that in this case the adversary cannot have a non-negligible advantage. By equation (5.2), then, it also cannot have a non-negligible advantage if the challenger uses credential j_b normally (i.e., without the help of the simulator $\mathcal{S}$). □

Theorem 9.16. *Our credential scheme is unlinkable.*

Proof. Follows from Theorems 9.14 and 9.15. □

Remark 9.17. Returning to Remark 7.13 on p. 140, notice that our scheme is unlinkable even though the Issue protocol is not (partially) blind (in the sense of Definition 7.1 on p. 127). For example, when given a credential $(k_0, \ldots, k_n), (\kappa, S, S_0, \ldots, S_n, T)$, the issuer can use k_0 and κ, and the numbers κ'' from the logs of Issue executions to recalculate the element R that would have been used during the Issue protocol, and try to match this R with the ones from his logs.

However, when combined with the ShowCredential protocol, this protocol still results in an (issuer and multi-show) unlinkable scheme. In this case, in contrast with the partially blind issuing protocol from Chapter 7, the unlinkability comes not from the Issue protocol but from the ShowCredential protocol: by being a zero-knowledge proof

of knowledge, the verifier learns nothing besides the fact that the user has a credential containing the disclosed attributes. Consequentially, even if the verifier was the one that issued the credential, the ShowCredential protocol hides so much that it cannot link ShowCredential executions to Issue executions.

9.3.3 Combining credentials using the private key

Let a bilinear pairing $e\colon G_1 \times G_2 \to G_T$ be fixed, and let PK and PK' be two public keys defined in these groups (not necessarily distinct). In this scenario multiple credentials can be bound together using their private keys k_0. If a user has a credential valid with respect to PK and another one valid with respect to PK' and the credentials have the same secret key, then the user can simultaneously disclose attributes from both credentials, in such a way that the verifier is assured that the two credentials are owned by one user, as opposed to two colluding users. This works as follows.

Let w and w' be the two credentials valid with respect to PK and PK', such that w and w' have the same secret key:

$$\begin{aligned} w &= ((k_0, k_1, \ldots, k_n), (K, S, S_0, \ldots, S_n, T)), \\ w' &= ((k_0, k'_1, \ldots, k'_n), (K', S', S'_0, \ldots, S'_n, T')). \end{aligned}$$

Now the user performs the ShowCredential credential once for each credential, but replaces the two proofs of knowledge by the following combined proof of knowledge:

$$\mathrm{PK}\Big\{ \big(k_0, \beta, \beta', \kappa, \kappa', (k_i)_{i\in\mathcal{C}}, (k'_j)_{j\in\mathcal{C}'}\big) : \\ D = \tilde{C}^{\beta} \bar{S}^{\kappa} \bar{S}_0^{k_0} \prod_{i\in\mathcal{C}} \bar{S}_i^{k_i} \ \wedge\ D' = (\tilde{C}')^{\beta'} (\bar{S}')^{\kappa'} (\bar{S}'_0)^{k_0} \prod_{i\in\mathcal{C}} (\bar{S}'_i)^{k'_i} \Big\}$$

This proves the same as the two separate proofs of knowledge, as well as the fact that the two credentials have the same secret key k_0.

This assumes that the user can control the private key during the issuing process, and this is indeed the case (see Figure 9.1). To ensure that each user has its own distinct secret key k_0, the issuing protocol could be modified as follows. In one version, the issuer also chooses a random $k''_0 \in_R \mathbb{Z}_p^*$, and then sets and sends

$$T = \Big(K S^{\kappa''} S_0^{k''_0} R \prod_{i=1}^{n} S_k^{k_i} \Big)^z$$

to the user together with κ'' and k''_0. The secret key of the new credential will then be not k_0 but $k_0 + k''_0$. In this way neither party can control the outcome of the secret key, and the issuer is prevented from learning its final value.

In the other version, the following happens:

- The user and issuer together decide in advance on a public key PK (that may or may not equal the issuer's public key), and a (possibly empty) disclosure set $\mathcal{D}$ and

set of attributes $(k_i)_{i\in\mathcal{D}}$).

- The user first performs the ShowCredential protocol, with the issuer acting as verifier, on a credential w that is valid with respect to PK, disclosing the attributes $(k_i)_{i\in\mathcal{D}}$. If successful, the user and issuer perform the Issue protocol as in Figure 9.1. However, the user combines the proof of knowledge over the hidden attributes with the proof of knowledge over k_0 and κ in the Issue protocol, additionally showing that k_0 coincides with the secret key from the credential that it showed.

9.4 Performance

9.4.1 The Fiat-Shamir heuristic

The zero-knowledge proof from Figure 5.3 on p. 100 that is used for proving knowledge of the numbers $\beta, \kappa, k_0, (k_i)_{i\in\mathcal{C}}$ consists of four moves. Although our scheme is thus far defined and proven secure in the standard model, if one is willing to assume the random oracle model, then we can apply the Fiat-Shamir heuristic [BR93; BPW12; FS87] to the Schnorr Σ-protocol for DL-representations (see Figure 5.2 on p. 98) as follows: the user receives a nonce $\eta \in_R \mathbb{Z}_p^*$ from the verifier, and uses the protocol from Figure 5.2 with $c = H(W, D, \eta)$ (for a suitable hash function H). It is easy to see that in the random oracle model, this 2-move protocol is a zero-knowledge proof of knowledge. This would not only lower the amount of exponentiations for the user in the ShowCredential protocol (with 6 in the case of the protocol from Figure 5.3), but also reduce the amount of moves to just two: after receiving the nonce η from the verifier, the user can combine the elements $\bar{K}, \bar{S}, \bar{S}_0, \ldots, \bar{S}_n, \tilde{C}, \tilde{T}$ and the proof of knowledge over η in a single message to the verifier (such a message is called a *disclosure proof*).

Concerns have been raised about the security of the random oracle model, however. For example, there exist protocols that are secure in the random oracle model, but do not have any secure standard model instantiation no matter which hash function is used [CGH04; GK03].

9.4.2 Exponentiation count

Although exponentiations (or scalar multiples, if we had written our groups additively) in elliptic curves are cheap compared to exponentiations in RSA groups, they are still the most expensive action that the user has to perform. In this section we will therefore count the number of exponentiations the user has to perform.

First note that

$$D = \bar{K}^{-1} \prod_{i\in\mathcal{D}} \bar{S}_i^{-k_i} = C^{-\alpha} \bar{S}^{\kappa} \bar{S}_0^{k_0} \prod_{i\in\mathcal{C}} \bar{S}_i^{k_i}.$$

These expressions for D contain $|\mathcal{D}|$ and $|\mathcal{C}| + 3$ exponentiations, respectively, so if $|\mathcal{C}| + 3 < |\mathcal{D}|$, the user should use the right hand side to determine D. Moreover, if the

user stores the elements $R := S^\kappa$ and $R_i := S_i^{k_i}$ for $i = 0, \ldots, n$, then it can calculate D by

$$D = \Big(K \prod_{i \in \mathcal{D}} R_i\Big)^{-\alpha} = \Big(C^{-1} R R_0 \prod_{i \in \mathcal{C}} R_i\Big)^{\alpha} \tag{9.9}$$

both of which take just one exponentiation.

Denote with $\mathrm{pk}(i)$ the amount of exponentiations the user has to compute in the zero-knowledge proof of knowledge when it presents a DL-representation of length i (in the case of the protocol from Figure 5.3 on p. 100, we have $\mathrm{pk}(i) = i + 6$, while $\mathrm{pk}(i) = i$ for the Fiat-Shamir heuristic applied to the Schnorr Σ-protocol). Then the number of exponentiations in G_1 that the user has to do is

- $n + |\mathcal{D}| + \mathrm{pk}(|\mathcal{C}| + 3) + 5$ exponentiations if $|\mathcal{D}| \leq |\mathcal{C}| + 3$,
- $n + |\mathcal{C}| + \mathrm{pk}(|\mathcal{C}| + 3) + 8$ exponentiations if $|\mathcal{D}| \geq |\mathcal{C}| + 3$,
- $n + \mathrm{pk}(|\mathcal{C}| + 3) + 6$ if the user stores and uses R and R_i for $i = 0, \ldots, n$.

The user performs no exponentiations in G_2 and G_T and computes no pairings. This results in great efficiency for the prover, since elements from G_2 are bigger and more expensive to deal with than those from G_1 (because generally $G_1 \subset E(\mathbb{F}_q)$ while $G_2 \subset E(\mathbb{F}_{q^k})[p]$, where k is the embedding degree of the curve).

Table 9.2 compares the amount of exponentiations in our scheme to those of [CL04], U-Prove and Idemix. However, note that exponentiations in RSA-like groups, on which Idemix depends, are significantly more expensive than exponentiations in elliptic curves. Also, the U-Prove showing protocol offers no unlinkability. As to the scheme from [CL04], Camenisch and Lysyanskaya did not include a showing protocol that allows attributes to be disclosed (that is, it is assumed that all attributes are kept secret), but it is not very difficult to keep track of how much less the user has to do if he voluntarily discloses some attributes. We see that the amount of exponentiations that the user has to perform in the ShowCredential protocol of [CL04] is roughly 1.5 times as large as in our scheme. Since, additionally, computing pairings is significantly more expensive than exponentiating, we expect our credential scheme to be at least twice as efficient.

9.4.3 Implementation

In order to further examine the efficiency of our credential scheme we have written a preliminary implementation, using the high-speed 254-bit BN-curve and pairing implementation from [Beu+10]. The latter is written in C++ and assembly but also offers a Java API, and it uses the GMP library from the GNU project[6] for large integer arithmetic. Table 9.3 shows the running times of our implementation along with those from the Idemix implementation from the IRMA project.[7] We have tried to make the comparison as honest as possible by writing our implementation in Java and using the Fiat-Shamir heuristic, like the IRMA Idemix implementation, which we have modified to also use the GMP library for its large integer arithmetic. However, the comparison can still only

[6] See `gmplib.org`.
[7] See `irmacard.org` and `github.com/credentials`.

Table 9.2. Exponentiation and pairing count for the user of the ShowCredential protocol of several attribute-based credential schemes. The columns G_{EC}, G_T and G_{RSA} show the amount of exponentiations in elliptic curves, the target group of a bilinear pairing, and RSA groups respectively, while the column labeled e counts the amount of pairings the user has to compute. The number n denotes the amount of attributes, excluding the secret key, and the function $\text{pk}(n)$ denotes the amount of exponentiations necessary in order to perform a zero-knowledge proof of knowledge of n numbers (in the case of the Fiat-Shamir heuristic applied to the Schnorr Σ-protocol, which Idemix also uses, we have $\text{pk}(n) = n$). In the case of our own scheme, we assume that the user calculates D in the ShowCredential protocol using the elements R_i as in equation (9.9).

	G_{EC}	G_T	e	G_{RSA}	unlinkable
Our scheme	$n + \text{pk}(\lvert\mathcal{C}\rvert + 3) + 6$	0	0	0	yes
[CL04]	$2n + 3$	$\text{pk}(\lvert\mathcal{C}\rvert + 2)$	$n + 3$	0	yes
Idemix	0	0	0	$\lvert\mathcal{C}\rvert + 3$	yes
U-Prove	$\lvert\mathcal{C}\rvert + 1$	0	0	0	no

go so far, because the elliptic curve group that [Beu+10] offers is heavily optimized for fast computations, from which our scheme profits because it allows multiple issuers to use the same group. Such optimizations are not possible in Idemix because each Idemix public key necessarily involves its own group. Moreover, the IRMA Idemix implementation is 1024-bits, which according to [LV01] corresponds to a 144 bit curve (see also `www.keylength.com`), so that the two implementations do not offer the same level of security.

For these reasons we will go no further than draw qualitative conclusions from the data. Nevertheless, both remarks actually demonstrate the efficiency of our scheme: the first means that our scheme can be optimized further than Idemix could, and Table 9.3 shows that even though our implementation offers a much higher level of security, it is still significantly faster than the IRMA Idemix implementation. We believe therefore that the conclusion that our scheme is or can be more efficient than Idemix – at least for the user in the ShowCredential protocol – is justified. Apart from this, the table also highlights the following differences between the two:

- In our scheme, verifying the validity of a disclosure proof tends to be two or three times as expensive as creating one, as it involves calculating a number of pairings. By contrast, in Idemix, verifying a disclosure proof is about as expensive as creating one.
- Verifying the validity of a credential in Idemix is significantly cheaper than computing or verifying disclosure proofs; while in our scheme verifying credential validity is only slightly cheaper than verifying disclosure proofs (again, because of having to compute pairings).
- When all but one attributes are disclosed (so that $\lvert\mathcal{C}\rvert = 1$ is constant), the ShowCredential protocol of our scheme becomes noticeably cheaper for the user. In Idemix, however, the user cost stops growing at all as the total amount of attributes in-

Table 9.3. A comparison of the running times of various actions in the implementation of our credential scheme and the IRMA Idemix implementation, both of them using the Fiat-Shamir heuristic. The columns labeled "computing proof" and "verifying proof" show how long it takes to compute and to verify a disclosure proof, respectively, while the column labeled "verifying credential" shows how long it takes to verify the signature of a credential. The left column shows the total number of attributes and, if applicable, the amount of disclosed attributes (this does not apply to the "verifying credential" column). In the case of our own scheme, we let the user calculate D in the ShowCredential protocol using the elements R_i as in equation (9.9). The attributes were randomly chosen 253-bit integers, the same across all tests, and the computations were performed on a dual-core 2.7 GHz Intel Core i5. All running times are in milliseconds, and were obtained by computing the average running time of 1000 iterations.

# attributes	computing proof		verifying proof		verifying credential	
total (discl.)	This work	Idemix	This work	Idemix	This work	Idemix
6 (1)	2.9	11.7	5.7	11.2	5.1	6.5
7 (1)	2.9	12.6	6.5	12.2	5.8	6.9
8 (1)	3.2	13.4	7.1	13.2	6.6	7.4
9 (1)	3.4	14.3	8.0	14.0	7.2	7.7
10 (1)	3.7	15.2	8.7	14.9	7.8	8.3
11 (1)	3.9	16.5	9.4	15.8	8.6	8.7
12 (1)	4.2	17.1	10.2	16.9	9.0	8.9
6 (5)	2.1	7.6	5.9	9.2		
7 (6)	2.1	7.5	6.5	9.7		
8 (7)	2.3	7.5	7.2	10.1		
9 (8)	2.4	7.4	7.9	10.7		
10 (9)	2.6	7.4	8.5	10.9		
11 (10)	2.7	7.5	9.1	11.4		
12 (11)	2.8	7.5	9.9	12.0		

creases. Referring to Table 9.2, this is because the amount of exponentiations in our scheme depends on n, while it does not in the case of Idemix. Verifying disclosure proofs becomes slightly cheaper in our scheme, but not by much because the amount of pairing stays the same; here too the differences are more pronounced in Idemix.

9.5 Proving unforgeability using the XKEA assumption

In this section we show how the whLRSW assumption from Definition 9.3 can be proven from (an extension of) the Known Exponent Assumption (KEA). Since the whLRSW assumption implies the unforgeability of our credential scheme, this results in a second unforgeability proof of our scheme.

9.5.1 The XKEA assumption

Let G be a cyclic group of prime order p, and let $1 \neq K \in G$. If an algorithm $\mathcal{A}$ is given K, K^a for some number $a \in \mathbb{Z}_p$, then it can generate a pair L, L^a by raising K and K^a to any power, i.e., by setting $(L, L^a) = (K^c, (K^a)^c)$ for any $c \in \mathbb{Z}_p$. The KEA assumption essentially states that this is the only way in which an algorithm can generate such a pair. There are many versions of this assumption, differing in how many such pairs the algorithm gets as input; if and how they are related; and if the algorithm is allowed to get auxiliary input besides these pairs.[8] The following version is the original one, introduced by Damgård in [Dam91]. We use the following notation: if $\mathcal{A}$ and $\mathcal{B}$ are two algorithms, then $X \leftarrow (\mathcal{A} \parallel \mathcal{B})(\sigma)$ indicates that $\mathcal{A}$ and $\mathcal{B}$ are given the same input σ and the random tape, and that X contains their concatenated output.

Definition 9.18 (KEA assumption). Let G be a cyclic group, whose prime order p is ℓ bits, and let $1 \neq K \in G$ and $a \in \mathbb{Z}_p^*$. The Knowledge of Exponent (KEA) assumption holds in G, if for any probabilistic polynomial-time algorithm $\mathcal{A}$ that receives input K, K^a, there exists a probabilistic polynomial-time algorithm $\chi_{\mathcal{A}}$ called the *extractor*, which is such that if $\mathcal{A}$ gives as output L, L', then $\chi_{\mathcal{A}}$, when given the same input and random coins of $\mathcal{A}$ returns $c \in \mathbb{Z}_p$ such that

$$\Pr[L' = L^a \ \wedge \ L \neq K^c] < \mathrm{negl}(\ell).$$

That is, more formally we have

$$\Pr\Big[a \in_R \mathbb{Z}_p^*;\ K \in_R G \setminus \{1\};\ \sigma \leftarrow (p, G, K, K^a); \quad (L, L', c) \leftarrow (\mathcal{A} \parallel \chi_{\mathcal{A}})(\sigma) : \\ L' = L^a \ \wedge \ L \neq K^c\Big] < \mathrm{negl}(\ell),$$

[8] There are also differences in the names given to the assumption: the KEA assumption is sometimes also called KEA1, SDHA-1 (for strong Diffie-Hellman), the KE assumption, or DHK (for Diffie-Helmann Knowledge). Meanwhile, the XKEA assumption is sometimes referred to as GKEA (for Generalized KEA), or n-KEA, or KEA3 when $n = 2$. The more recent papers seem to have settled on some variant of KEA.

where the probability is over the selection of a and K, and the randomness used by $\mathcal{A}$ and $\chi_{\mathcal{A}}$.

The phrasing of the assumption leaves open the possibility that the output of $\mathcal{A}$ does not have a as the relative discrete log (for example, $\mathcal{A}$ may output that it failed). When it does, however, output a pair having a as relative discrete log, then the probability that the extractor does not manage to find the discrete log of L with respect to K is negligible.

The KEA assumption and its various generalizations are used in areas such as (non-interactive) zero-knowledge proofs [BP04a; Bit+14; Bit+12; Gro10], encryption [BP04b; Dam91; WS08], 2-party computation [DFH12], and authentication and key-exchange [DG09; DGK06; Kra05; WS07; YZ10]. It has been criticized because it seems harder to disprove it than assumptions such as the discrete logarithm assumption, in the sense that it is not falsifiable [Nao03]. On the other hand, like the LRSW assumption it can be proven to hold in the *generic group model*, in which the adversary is oblivious of the actual representation of the group elements and only performs generic group operations such as multiplying, inverting, and testing for equality. In for example [Gro10] this is proved even in the presence of bilinear pairings. We briefly expand on this in the next subsection, but first we present the version of the assumption that we need.

This version is the same as that of, for example, [Bit+12] (although ours is less permissive in the auxiliary input to the adversary). It generalizes the assumption above by allowing $\mathcal{A}$ to receive more than one pair of the form (K, K^a), and to a Type 3 pairing setting.

Definition 9.19 (XKEA assumption). Let $G_1 \times G_2 \to G_T$ be a Type 3 bilinear pairing, where the order p of the three groups is ℓ bits, and let and $a \in \mathbb{Z}_p^*$. Let $\mathcal{A}$ be a probabilistic polynomial-time algorithm taking as input $K_1, \ldots, K_m, K_1^a, \ldots, K_m^a \in G_1$, $Q_1, \ldots, Q_r \in G_2$, where

- m and r are polynomial in ℓ,
- the $K_1, \ldots, K_m$ are uniformly randomly distributed while the $Q_1, \ldots, Q_r$ may be arbitrarily distributed (i.e., we do not impose any restriction on the distribution of the $Q_1, \ldots, Q_r$),
- $K_j \neq 1$ for all $j = 1, \ldots, n$.

The Extended Knowledge of Exponent (XKEA) assumption holds in G, if for any such $\mathcal{A}$ there exists a probabilistic polynomial-time algorithm $\chi_{\mathcal{A}}$ called the *extractor* which is such that if $\mathcal{A}$ gives as output L, L', then $\chi_{\mathcal{A}}$, when given the same input and random coins of $\mathcal{A}$ returns a tuple $c_1, \ldots, c_n$ such that

$$\Pr[L' = L^a \ \wedge \ L \neq K_1^{c_1} \cdots K_m^{c_m}] < \mathrm{negl}(\ell).$$

More formally, we have for any tuple $Q_1, \ldots, Q_r$ of polynomial length r in ℓ,

$$\begin{aligned} \Pr \Big[& a \in_R \mathbb{Z}_p^*;\ K_1, \ldots, K_m \in_R G \setminus \{1\}; \\ & \sigma \leftarrow (p, e, G_1, G_2, G_T, K_1, \ldots, K_m, K_1^a, \ldots, K_m^a, Q_1, \ldots, Q_r); \\ & (L, L', c_1, \ldots, c_m) \leftarrow (\mathcal{A} \,\|\, \chi_{\mathcal{A}})(\sigma) : \quad L' = L^a \ \wedge \ L \neq K_1^{c_1} \cdots K_m^{c_m} \Big] < \mathrm{negl}(\ell), \end{aligned} \tag{9.10}$$

where the probability is over the selection of $a, K_1, \ldots, K_m$ and the randomness used by $\mathcal{A}$ and χ_A.

Notice that the KEA assumption does not imply the difficulty of the discrete logarithm problem: in groups where the DL problem is easy, if $(L, L^a) \leftarrow \mathcal{A}(K, K^a)$ then the extractor that computes and returns $\log_K L$ would be efficient. Thus, KEA can (and will) also hold in groups in which DL is easy. For the same reason the XKEA assumption does not imply the BDL assumption (see Section 9.2.2), so that we will have to separately take the BDL assumption.

9.5.2 The XKEA assumption and the generic group model

The difficulty of solving a computational problem depends on how it is represented. Consider for example the Decisional Diffie Hellman problem (DDH, see Definition 5.10 or Example 5.13). In $\mathbb{Z}_p$ this is easy, but there exist many elliptic curves over $\mathbb{Z}_p$ for which DDH is believed to be hard, even though the two groups are isomorphic (the isomorphism being $a \mapsto P^a$ for any element $P \neq 1$). The generic group model is a way of studying algorithms that cannot exploit the representation of group elements, but only use the group structure.

In the context of the KEA assumption, this notion is usually formalized using techniques proposed by Shoup [Sho97] (where it is also proven that the discrete logarithm problem is hard in the generic group model). In this model, algorithms are not provided direct access to group elements, but only to the images of group elements of a random bijection $\sigma\colon \mathbb{Z}_p \to \mathbb{Z}_p$, together with oracle access to the following two functions:

$$\mathsf{add}(\sigma(x), \sigma(y)) = \sigma(x + y),$$
$$\mathsf{inv}(\sigma(x)) = \sigma(-x).$$

This also provides the ability to take random elements to the algorithm, by querying the oracle on $\sigma(x')$ for any previously unseen x'. If there is a bilinear pairing, this can also be modeled using such an oracle function. It is clear that in this situation, the attacker can gain no advantage in solving a computational problem using the representation $\sigma(x)$ of an element x. In the case of the KEA assumption, if $(L, L^a) \leftarrow \mathcal{A}(K, K^a)$, one can show that with overwhelming probability the number c such that $L = K^c$ can be calculated by observing the oracle calls made by $\mathcal{A}$.

Let n denote the amount of pairs $(K_i, S_i = K_i^a)$ that our algorithm receives. The first two proofs along these lines seem to be from [Den06], where it is proved for $n = 1$, and [AF07], where it is proved for $n = 1, 2$ in the presence of a symmetric (i.e. Type 1) pairing. The latter proof can easily be modified into one that allows

- a Type 3 pairing instead of Type 1,
- $n > 2$ pairs of elements K_i, K_i^a,
- any set of elements from G_2 as auxiliary input,[9]

[9]Some papers allow the algorithm to additionally receive *arbitrary* auxiliary input $\xi \in \{0,1\}^*$. We are more careful than that, because the statement in this form could impossibly hold for any group in which the discrete logarithm problem is hard: if $\xi = a$, then from the program that takes a random $C \in G_1$ and returns C, C^a, no coefficient can be extracted (besides a).

resulting in the version from Definition 9.19.

9.5.3 Unforgeability based on XKEA

First we need the following lemma. Recall that Propositions 5.7 on p. 89 and 9.6 on p. 157 state that when the (bilinear) discrete logarithm problem is hard, then no probabilistic polynomial-time algorithm can create non-trival DL-representations of 1, with respect to a set of uniformly randomly distributed base points. In the lemma below, half of the base points are randomly distributed, while the other half of the base points is a fixed power of the first half. It is easy to extend the proof of Proposition 5.7 to this case.

Lemma 9.20. *Let $e\colon G_1 \times G_2 \to G_T$ be a Type 3 pairing in which the BDL assumption holds, let p be the order of the three groups, and let P, Q be generators of G_1, G_2 respectively. Let the tuple $X_1, \ldots, X_m \in G_1$ and the tuple $Y_1, \ldots, Y_m \in G_2$ be such that $\log_P X_j = \log_Q Y_j$, i.e., $e(P, Y_j) = e(X_j, Q)$.*

In addition, suppose that m is even, and that

- *$X_1, \ldots, X_{m/2} \in G_1$ are randomly distributed,*
- *$X_{j+m/2} = X_j^a$ for some $a \in \mathbb{Z}_p^*$ and all $j = 1, \ldots, m/2$ (and similarly $Y_{j+m/2} = Y_j^a$).*

Then no probabilistic polynomial-time algorithm can, on input $X_1, \ldots, X_m, Y_1, \ldots, Y_m$ generate a non-trivial DL-representation of $1 \in G_1$ with respect to $(X_1, \ldots, X_m)$.

Theorem 9.21. *The XKEA and BDL assumptions imply the whLRSW assumption, as well as the unforgeability of our credential scheme from Section 9.3 and our signature scheme from Section 9.2.3.*

Proof. Since the whLRSW assumption implies the unforgeability of our credential scheme, the second part of the statement is clear. Suppose then that we have an algorithm $\mathcal{A}$ that violates the whLRSW assumption, while the XKEA and BDL assumptions hold. That is, algorithm $\mathcal{A}$ is given $(p, e, G_1, G_2, G_T, Q, Q^a, Q^z)$, along with a list of LRSW-instances $(\kappa_j, K_j, K_j^a, K_j^{z+\kappa_j az})$ of polynomial length, and it outputs an LRSW-instance with a new κ with non-negligible probability. We show that as a consequence of the XKEA assumption, we can extract a set of numbers from $\mathcal{A}$ that will result in a non-trivial DL-representation of 1. This will contradict the lemma above.

First, we must justify an application of the XKEA assumption to Algorithm $\mathcal{A}$. We know that $\mathcal{A}$ outputs K, S such that $S = K^a$ with non-negligible probability. However, $\mathcal{A}$ needs elements $\kappa_j, T_j = K_j^{z+\kappa_j az}$ as input, besides the elements K_j, S_j (which indeed have a as relative discrete log), while the XKEA assumption allows only elements from G_2 as extra input (see Definition 9.19). We now define a second algorithm $\mathcal{B}$ to which the XKEA assumption does apply as above. (For the moment, the fact that $\mathcal{A}$ should also output κ, T such that $T = K^{z+\kappa az}$ is not important, so we momentarily consider them thrown away from the output of $\mathcal{A}$.) Suppose that there exists no extractor for $\mathcal{A}$ – that is, all algorithms are not an extractor for $\mathcal{A}$: for all probabilistic polynomial-time

Algorithm 1 Algorithm $\mathcal{B}$.

function $\mathcal{B}(p, e, G_1, G_2, G_T, \{K_j, S_j\}_j, Q, A)$
  $z \in_R \mathbb{Z}_p^*$
  $Z \leftarrow Q^z$
  for all $j \in \{1, \ldots, m\}$ **do**
    $\kappa_j \in_R \mathbb{Z}_p^*$
    $T_j \leftarrow (K_j S_j^{\kappa_j})^z$
  end for
  return $\mathcal{A}(p, e, G_1, G_2, G_T, \{\kappa_j, K_j, S_j, T_j\}_j, Q, A, Z)$
end function

algorithms χ, we must have

$$\Pr\Big[a, z, \kappa_1, \ldots, \kappa_m \in_R \mathbb{Z}_p^*;\ K_1, \ldots, K_m \in_R G_1 \setminus \{1\};\\ Q \in_R G_2;\ A \leftarrow Q^a;\ Z \leftarrow Q^z; \quad \{T_j \leftarrow K_j^{z+\kappa_j a z}\}_j;\\ \sigma \leftarrow (p, e, G_1, G_2, G_T, \{\kappa_j, K_j, K_j^a, T_j\}_j, Q, A, Z);\\ (K, S, c_1, \ldots, c_m) \leftarrow (\mathcal{A} \,\|\, \chi)(\sigma) :\\ \qquad S = K^a \ \wedge\ K \neq K_1^{c_1} \cdots K_m^{c_m}\Big] > \mathrm{negl}(\ell), \tag{9.11}$$

This says that the probability that our would-be extractor χ does *not* output coefficients $c_1, \ldots, c_m$ such that $K = K_1^{c_1} \cdots K_m^{c_m}$ is not negligible; i.e., for all χ there is a significant probability that it fails as an extractor for $\mathcal{A}$. Now notice that the only thing algorithm $\mathcal{B}$ does, besides executing algorithm $\mathcal{A}$, is choosing the κ_j and z, and calculating Z and the T_j exactly as in the experiment in the first two lines of the probability above. This means that using $\mathcal{B}$ we can write the probability above more concisely as follows: for all probabilistic polynomial-time algorithms χ we have

$$\Pr\Big[a \in_R \mathbb{Z}_p^*;\ K_1, \ldots, K_m \in_R G_1 \setminus \{1\};\ Q \in_R G_2,\ A \leftarrow Q^a;\\ \sigma \leftarrow (p, e, G_1, G_2, G_T, Q, A, K_1, \ldots, K_m, K_1^a, \ldots, K_m^a);\\ (K, S, c_1, \ldots, c_m) \leftarrow (\mathcal{B} \,\|\, \chi)(\sigma) :\\ \qquad S = K^a \ \wedge\ K \neq K_1^{c_1} \cdots K_m^{c_m}\Big] > \mathrm{negl}(\ell).$$

This directly contradicts equation (9.10). That is, this would imply that there exists no XKEA-extractor for $\mathcal{B}$, which cannot be by assumption. Therefore, our assumption (9.11) cannot hold, meaning that there is in fact an extractor for $\mathcal{A}$ that can extract the numbers $c_1, \ldots, c_n$. Notice that this extractor does not need to know or be given z as an argument, even though algorithm $\mathcal{B}$ does know z.

The application of the XKEA assumption to Algorithm $\mathcal{A}$ relative to the secret key z can be justified in the same way. Algorithm $\mathcal{B}$ now receives C_i, T_i for $i = 1, \ldots, m$,

generates $a \in_R \mathbb{Z}_p^*$ and does the following in its for-loop:

$$\kappa_i \in_R \mathbb{Z}_p^*$$
$$K_i \leftarrow C_i^{1/(1+a\kappa_i)}$$
$$S_i \leftarrow K_i^a.$$

Then $T_i = C_i^z = (K_i S_i^{\kappa_i})^z$ as required, so that it can invoke algorithm $\mathcal{A}$ as above.

Thus, we may conclude that we can indeed extract numbers $c_1, \ldots, c_m$ and $d_1, \ldots, d_m$ from algorithm $\mathcal{A}$ which are such that

$$K = \prod_{j=1}^{m} K_j^{c_j}, \qquad S = \prod_{j=1}^{m} S_j^{c_j},$$
$$T = \prod_{j=1}^{m} T_j^{d_j}, \qquad C = \prod_{j=1}^{m} C_j^{d_j} = \prod_{j=1}^{m} K_j^{d_j} S_j^{\kappa_j d_j}.$$

On the other hand, notice that C should be equal to

$$C = KS^{\kappa} = \prod_{j=1}^{m} K_j^{c_j} S_j^{\kappa c_j}.$$

Comparing the right hand sides of the two equations gives

$$\prod_{j=1}^{m} K_j^{d_j} S_j^{\kappa_j d_j} = \prod_{j=1}^{m} K_j^{c_j} S_j^{\kappa c_j}.$$

This results in a DL-representation of 1, namely

$$1 = \prod_{j=1}^{m} K_j^{d_j - c_j} S_j^{\kappa_j d_j - \kappa c_j}. \tag{9.12}$$

Next, we show that the numbers

$$d_j - c_j \qquad \kappa_j d_j - \kappa c_j \tag{9.13}$$

are not all equal to 0, so that (9.12) is a nontrivial representation of 1. Suppose they do all equal 0. This gives the following set of equations for all j:

$$d_j = c_j, \qquad \kappa c_j = \kappa_j d_j. \tag{9.14}$$

These equations imply that for each j, we either have $(c_j, d_j) = (0, 0)$, or $\kappa = \kappa_j$. There must exist at least one pair (c_j, d_j) unequal to $(0, 0)$, otherwise we would have $C = 1$ which is not valid. Therefore, this contradicts the assumption that the adversary output

a credential with a new κ.

Thus the numbers (9.13) do not all equal 0, so that the DL-representation of 1 in equation (9.12) is nontrivial. Now if we let algorithm $\mathcal{B}$ defined above extract the numbers c_j, d_j from $\mathcal{A}$, then it can efficiently compute and return the DL-representation (9.12), contradicting Proposition 9.20. □

9.6 Conclusion

In this chapter we have defined a new self-blindable attribute-based credential scheme, and given a full security proof in the standard model by showing that it is unforgeable and unlinkable. The unforgeability of our scheme can be proven using a standard hardness assumption as well as through the Known Exponent assumption. Based on the fact that it uses elliptic curves and bilinear pairings (but the latter only on the verifier's side), on a comparison of exponentiation counts, and on a comparison of run times with the IRMA Idemix implementation, we have shown it to be more efficient than comparable schemes such as Idemix and the scheme from [CL04], achieving the same security goals at less cost.

Although our scheme is unforgeable and self-blindable, it is not susceptible to the problem that Theorem 6.6 on p. 118 expresses. Apart from the fact that the underlying signature scheme is nondeterministic while it is deterministic in each of the schemes that we studied in Chapter 6, in our ShowCredential protocol there is no public key that is sent to the verifier separate from the signature and attributes. Since an application of Theorem 6.6 critically depends on such public keys, it does not apply.

A significant difference between our scheme from this chapter (as well as the scheme from [CL04]) on the one hand, and Idemix, U-Prove and our (linkable) scheme from Chapter 7 on the other hand, is that the length of the signature grows with the number of attributes n. The signature length of the forgeable scheme from Chapter 8 is also constant. It is interesting to note that the reason that our scheme is not vulnerable to the attack from that chapter is precisely that each credential in our scheme has its own set of base points, which is what makes the signature length linear in n as opposed to constant. These observations suggest that a scheme with fixed base points and known prime group order cannot result in a scheme which is simultaneously unlinkable and unforgeable. Intuitively, the relatively simple structure of the group seems to offer too much freedom to modify the credentials. In that sense, our scheme from this chapter may be the best that one can do using (prime-order) elliptic curve groups.

End Matter

Chapter 10

Summary and conclusions

10.1 Quantization using jet space geometry

Starting from the classical theory of a point particle on a smooth manifold, we have seen two directions in which this can be generalized. In Chapter 1 we studied the variational Schouten bracket on jet spaces, which specializes to the ordinary Schouten bracket on manifolds. This bracket is an important part of the BV-formalism that we studied in Chapter 2. Both of these chapters revolved around secondary calculus: the idea that in generalized theories, the geometric machinery should have the same physical meaning and role as they do in specialized theories. This attractive idea not only highlights connections between mathematics and the physics that it can describe, but also results in interesting relations and results purely within mathematics, while simultaneously bringing more clarity and preciseness in the mathematical description of physical theories.

Chapter 3 considered a generalization of the point particle in a manifold in an entirely different direction, namely quantum mechanics. We saw that in the case of a Poisson manifold there exists a star product on the ring of functions of the manifold, that unites the pointwise product, the Poisson structure, and the dynamics of the particle in a single object. We included an explicit description of this star product in the case of the duals of Lie algebras, which carry a natural Poisson structure.

When considering generalizations of a notion or theory, one question that can be very difficult to answer is in which direction it should be generalized. There are often many candidates, that sometimes turn out to be the same (such as the multiple definitions of the variational Schouten bracket from Chapter 1), while others turn out not to satisfy all requirements. In Chapter 4 we examined three possible variational generalizations of star products, all of which turned out to be unsuitable. Such a star product would simultaneously generalize Chapters 1 and 2 on the one hand, and Chapter 3 on the other

hand. Physically, it could lead to a quantum mechanical theory of very general physical configurations, so the question if a suitable variational star product exists and what it looks like remains very interesting. Although the direction that we took did not lead to success, we remain convinced that the setting of infinite jet bundles combined with the framework of deformation quantization can lead to theories that are of great importance to both mathematics and physics.

10.2 Identity management using credential schemes

Elliptic curves and bilinear pairings offer a remarkable combination of security, efficiency and flexibility. In particular, the fact that the order of the groups involved is prime and known to all participants of the scheme under consideration results in a rather simple group structure, when compared to for example groups of composite order that are used by RSA and Idemix. The difference between the two mathematical settings is somewhat comparable to the differences between fields and rings. (Actually, this is no coincidence: when raising group elements of a cyclic group G to a power, one uses the natural module structure of the group over its own ring of coefficients $\mathbb{Z}/|G|\mathbb{Z}$. In the case of (subgroups of) elliptic curves of prime order p, however, the ring of coefficients $\mathbb{Z}/p\mathbb{Z}$ is indeed a field, so that the curve becomes a 1-dimensional vector space over this field. This is not the case for groups whose order is not a prime power, such as for example the group that Idemix uses; in fact, such groups can impossibly be a vector space over any field.)

At the same time, this simple group structure endows all participants with more ways to manipulate and relate the various group elements, so that it can be difficult to prevent adversarial parties from breaking the security or anonymity objectives of the scheme. We have seen this in Chapter 6, which showed that in certain circumstances there is a trade-off between unforgeability and anonymity, in the form of unlinkability; and more dramatically in Chapter 8, where we showed that a proposed attribute-based scheme that was meant to unite high efficiency, unforgeability and unlinkability is not in fact unforgeable.

We have examined a number of credential schemes in this part of the thesis which progressively attained more of the four goals stated in the Introduction on p. xiv, namely security, unlinkability, efficiency and support for attributes. In Chapter 7 we introduced a provably secure attribute-based scheme in which the length of the credentials is constant, resulting in great efficiency. However, the scheme did not provide unlinkability. The unlinkable scheme from Chapter 8 also has credentials of constant length, but we found that it is forgeable. Finally, in Chapter 9 we introduced an unforgeable and unlinkable attribute-based credential scheme that demonstrates that it is in fact possible to unite efficiency with unforgeability and anonymity (in the sense of issuer and multi-show unlinkability) in this mathematical background. This scheme is not as efficient as the linkable scheme from Chapter 7, highlighting that there indeed exists a trade-off between efficiency versus anonymity. However, our scheme is the most efficient when compared with all other unlinkable schemes that we know of, and it should be implementable on smart cards.

In conclusion, to date there seems to be no unlinkable attribute-based credential

scheme using elliptic curves in which the size of the signature of a credential not depend on the amount of attributes. Although we have reason to believe that such a scheme may not be possible at all, this would be a definite improvement on the scheme from Chapter 9. For that reason, a scheme offering those features, or a proof that it does not exist would be of great interest.

Inleiding en samenvatting

Dit hoofdstuk is een Nederlandse versie van de introductie en samenvatting die begint op pagina vii.

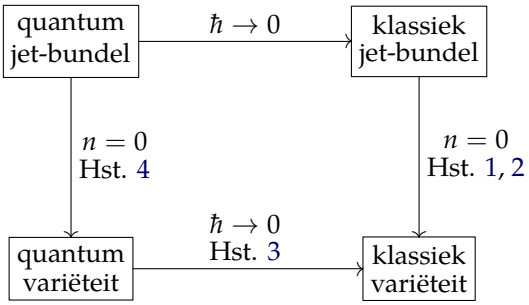
Quantisatie met behulp van jet-bundels

Introductie

Als er iets is wat wiskundigen en natuurkundigen gemeen hebben, dan is het hun waardering voor esthetiek: zowel de wiskundige als de natuurkundige streeft naar dat perfecte bewijs of die elegante theorie. Hoewel ze het er niet altijd over eens zijn wat mooi is en wat niet, kunnen ze het in de overlappende gebieden vaak goed met elkaar vinden. Dit kan leiden tot, en heeft ook geleid tot, interessante kruisbestuiving.

Het eerste deel van dit proefschrift draait om een thema wat door mensen uit beide kampen gewaardeerd wordt, namelijk specialisatie: het toepassen van een algemene theorie op een welbekende, specifieke situatie. Het is vaak echter interessanter om te proberen de andere kant op te gaan, en te proberen om een specifieke theorie te veralgemeniseren naar meer situaties en toepassingen. Dit kan een constructief en inspirerend principe zijn in het zoeken naar en het begrijpen van nieuwe theorieën. In beide vakgebieden heeft dit pad geleid tot steeds algemenere en krachtigere theorieën. Dit leidt tot de volgende vraag die in dit eerste deel een grote rol speelt: *Hoe kunnen we de concepten veralgemenisering en specialisatie gebruiken om ons begrip van de verbanden tussen wiskunde en theoretische natuurkunde uit te breiden, in het bijzonder binnen de geometrie van jet-bundels en quantisatiemethoden, en tot in hoeverre kan dit leiden tot een vooruitgang in de wiskundige zijden van deze verbanden?*

Een eerste voorbeeld hiervan is te vinden in hoofdstukken 1 en 2. De meeste partiële differentiaalvergelijkingen die men vindt in natuurkunde, en ook veel die afkomstig zijn uit de wiskunde, kunnen op elegante wijze worden beschreven in termen van oneindige jet-bundels. De onafhankelijke variabelen vormen samen de basisruimte, terwijl



Figuur 1. Een schematisch overzicht van de hoofdstukken en hun onderwerp. De tekst in de vertices geeft aan of de theorie quantummechanisch of klassiek is, en of ze gaat over gladde variëteiten of jet-bundels. Een pijl geeft aan dat men de theorie aan het begin van de pijl kan specialiseren naar de theorie aan het eind van de pijl, door de limiet te nemen die boven of naast de pijl staat (n is de dimensie van de basisruimte). Merk op dat de hoofdstukken in feite de andere kant opgaan.

de onbekendes de vezel vormen van een vezelbundel boven deze basisruimte. De differentiaalvergelijking zelf is dan een deelvariëteit van de oneindige jet-bundel van de vezelbundel. Binnen de natuurkunde kan men hiermee bijvoorbeeld de dynamica van velden, golven en snaren beschrijven. Echter, wanneer men deze machinerie toepast op een puntdeeltje wordt de partiële differentiaalvergelijking een gewone differentiaalvergelijking, wordt de basisruimte een enkel punt, en wordt de oneindige jet-bundel een (eindig-dimensionale) gladde variëteit. De natuurkunde van zo'n puntdeeltje is welbekend, en wordt geformuleerd in termen van de geometrische structuren rond variëteiten, zoals vectorvelden en de de Rham-differentiaal. Onder de "specialisatiefunctie" die een algemene theorie stuurt naar het puntdeeltje kunnen we ons dan afvragen wat de inverse van deze geometrische machinerie is. Het idee dat zulke inverses bestaan, en dat ze in de algemene theorie ongeveer dezelfde rol moeten spelen als in het speciale geval, is daarmee de wiskundige versie van het idee dat de natuurkunde van een puntdeeltje een speciaal geval is van algemenere theorieën. Dit idee wordt secundaire calculus genoemd. Hoewel het in het hele eerste deel van dit proefschrift een grote rol speelt, is het het meest aanwezig in hoofdstuk 2 (zie sectie 2.2 op p. 26).

In hoofdstuk 3 komt een tweede voorbeeld voor. In de natuurkunde draait de quantummechanica om de constante van Planck, $\hbar \approx 1.055 \times 10^{-34}$ Js. Quantummechanische theorieën zijn doorgaans zodanig dat (als we de constante aard van de constante van Planck tijdelijk negeren) het nemen van de limiet $\hbar \to 0$ resulteert in een klassieke theorie, die een benadering is van de quantummechanische. Wiskundig ligt het daarom voor de hand om $\hbar$ te beschouwen als een deformatieparameter, zodat de quantummechanische theorie een deformatie van de klassieke theorie is. Dit is in essentie het idee van deformatie-quantisatie. Daarin begint men met een klassiek theorie van een puntdeeltje dat zich bevindt in een zekere variëteit M. De dynamica van het deeltje wordt beschreven door een Poisson-structuur en een Hamiltoniaans vectorveld op M. Natuurkundig gezien is de ring van functies $C^\infty(M)$ dan de ruimte van observabelen. De quantummechanische ruimte van observabelen is dan een deformatie in $\hbar$ van $C^\infty(M)$;

dit leidt tot een non-commutatief product op de machtreeksen over $\hbar$ van $C^\infty(M)$, die het ster-product genoemd wordt. Dit product bevat op een bepaalde manier niet alleen het oorspronkelijke puntsgewijze product op $C^\infty(M)$ maar ook de Poisson-structuur in termen waarvan de dynamica wordt beschreven. Dit leidt tot een elegante quantumtheorie van puntdeeltjes, die inderdaad zodanig is dat men de klassieke theorie terugkrijgt in de limiet $\hbar \to 0$. (Deze methode van klassieke theorieën quantiseren is zeker niet de enige; in de natuurkundige literatuur worden de BRST- en BV-formalismes een stuk meer gebruikt, aangezien zij een belangrijke rol spelen in het zeer succesvolle standaardmodel).

Figuur 1 vat de verbanden tussen de hoofdstukken samen. We zien dat zowel hoofdstuk 2 als hoofdstuk 3 beginnen bij een klassieke theorie van een puntdeeltje op een variëteit, en dat dan generaliseren naar andere onderwerpen dan puntdeeltjes (dus naar jet-bundels), en tot een quantumtheorie (dus naar quantisatie-deformaties), respectievelijk. Dit leidt aan de ene kant tot een klassieke theorie die vele natuurkundige fenomenen kan beschrijven, en aan de andere kant tot een quantumtheorie van puntdeeltjes. Een voor de hand liggende vraag is dan of deze twee uitersten beiden specialisaties zijn van een quantumtheorie van algemenere natuurkundige configuraties dan puntdeeltjes – dat wil zeggen, van de theorie die zich zou bevinden in de linker bovenhoek van figuur 1. Wiskundig gezien zou dit dan een theorie over deformatie-quantisaties op jet-bundels zijn. In hoofdstuk 4 gaan we kort op deze vraag in.

Samenvatting

Het eerste deel van het proefschrift bevat de volgende hoofdstukken.

Hoodstuk 1 introduceert eerst kort oneindige jet-bundels en een aantal gerelateerde constructies, aangezien deze een grote rol spelen in het hele eerste deel. Daarna beschrijven we multivectoren op jet-bundels als een eerste toepassing van secundaire calculus. Deze vormen het domein van de variationele Schouten-haak, die twee van zulke multivectoren neemt en een nieuwe teruggeeft; we beschrijven hoe deze haak een natuurlijke generalisatie is van de Schouten-haak op gladde variëteiten. De variationele Schouten-haak is in de afgelopen decennia in verschillende gedaantes voorgekomen in de wiskundige en natuurkundige literatuur, wat heeft geleid tot een aantal verschillende beschrijvingen van hetzelfde concept. In de laatste sectie van dit hoofdstuk laten we zien hoe deze verschillende versies equivalent zijn. Dit hoofdstuk is gebaseerd op het volgende artikel.

> [KR12] A. V. Kiselev en S. Ringers. "A comparison of definitions for the Schouten bracket on jet spaces". In: *Proceedings of the Sixth International Workshop "Group Analysis of Differential Equations and Integrable Systems"*. Larnaca, Cyprus, 2012, 15p. arXiv: 1208.6196.

Hoofdstuk 2 draait om het Batalin-Vilkovisky formalisme, een techniek voor het quantiseren van klassieke natuurkundige theorieën. Samen met de Schouten-haak uit het vorige hoofdstuk is een belangrijk ingredient van dit formalisme de BV-Laplaciaan: een differentiaaloperator van orde twee die kwadrateert naar 0, en die samen met

de Schouten-haak een wiskundige structuur genaamd BV-algebra moet vormen. Een probleem met deze BV-Laplaciaan zoals hij doorgaans gedefinieerd wordt, echter, is dat er vaak "oneindige constantes" of deltafuncties in voorkomen, die dan verwijderd moeten worden door middel van regularisatie. Na de meetkunde te hebben beschreven in termen van oneindige jet-bundels geïntroduceerd in het vorige hoofdstuk, verkennen we kort een mogelijke BV-Laplaciaan die geen oneindige constantes heeft. Echter, we zullen zien dat de voor de hand liggende manier om dit te doen niet leidt tot een BV-algebra.

Hoofdstuk 3 draait om deformatie-quantisatie, wat een heel andere techniek is voor het quantiseren van natuurkundetheorieën (maar alleen voor puntdeeltjes). We beschrijven eerst kort het klassieke natuurkundige formalisme van puntdeeltjes in termen van Poisson-structuren en Hamiltoniaanse vectorvelden op gladde variëteiten. Door middel van Kontsevich' beroemde resultaat over deformatie-quantisatie van Poisson-variëteiten kan zo'n theorie altijd worden geconverteerd tot een quantummechanische theorie, waarvan het belangrijkste ingrediënt een niet-commutatief maar associatief product is, die het sterproduct genoemd wordt. Na dit concept te hebben gedefinieerd en Kontsevich' resultaat kort te hebben beschreven, geven we een grondige uitwerking van dit formalisme toegepast op de natuurlijke Poisson-haak van de duale van een Lie-algebra.

Hoofdstuk 4 bestudeert kort een aantal manieren om secundaire calculus in de vorm van oneindige jet-bundels te combineren met deformatie-quantisatie. Poisson-structuren en Hamiltoniaanse vectorvelden hebben in het raamwerk van secundaire calculus generalisaties naar oneindige jet-bundels. We kunnen ons dan afvragen of er zoiets is als variationele deformatie-quantisaties: een generalisatie van sterproducten naar jet-bundels. Natuurkundig gezien zou zo'n theorie een generalisatie kunnen zijn van de methode die puntdeeltjes generaliseert naar meer algemene structuren door middel van deformatie-quantisatie. Er zijn meerdere mogelijke manieren waarop men zo'n generalisatie kan proberen, waarvan we in dit laatste hoofdstuk laten zien dat een aantal van deze manieren niet werken.

Identiteitsbeheer met behulp van credentialschema's

Introductie

In de afgelopen jaren is het belang van informatie in onze maatschappij enorm snel toegenomen, vooral dankzij de introductie en alomtegenwoordigheid van computers en het internet. Steeds meer van de dingen die we doen gebeurt online. Mensen gebruiken bijvoorbeeld het internet om in contact te blijven met elkaar, om nieuwe mensen te ontmoeten, om dingen te kopen of verkopen, om afspraken te maken, en om hun belastingopgave in te dienen. Bij al deze acties wordt informatie gegenereerd, die kan worden opgeslagen en geanalyseerd. Daarnaast heeft sinds een paar jaar bijna iedereen constant een *smart phone* bij zich, die meestal continu verbonden is met het internet. Hiermee kunnen allerlei verschillende soorten data ten eerste continu worden geregistreerd – bijvoorbeeld, geluid, locatie en beweging, maar ook waar je op klikt, wat je koopt en met wie je in contact staat – en ten tweede kan deze data snel en efficiënt

over het internet worden verzonden. Tegelijkertijd nemen de kosten van het verzenden en opslaan van data alsmaar af, en moderne databases kunnen gemakkelijk enorme hoeveelheden data opslaan en daar snel toegang toe bieden.

Informatie over eigenschappen en handelingen van individuen, kortweg *persoonlijke informatie*, is met name belangrijk. In mijn geval zijn mijn leeftijd, naam en woonplaats voorbeelden hiervan, maar ook dat ik hou van klassieke muziek, dat ik een aantal weken terug een e-book heb gekocht, en ook dat ik een relatie heb. Eerder verrichte handelingen behoren ook tot persoonlijke informatie, samen met bijbehorende gerelateerde informatie zoals gesproken of geschreven tekst, financiële gegevens, of implicaties zoals dat ik geïnteresseerd ben in cryptografie.

Gegeven het belang van dergelijke informatie en het gemak waarmee het tegenwoordig kan worden verzameld is het geen verrassing dat verschillende partijen het zijn gaan verzamelen, op variërende manieren en voor verschillende doelen. Bijvoorbeeld, in essentie is het verdienmodel van zowel Facebook als Google het verzamelen van persoonlijke informatie van mensen om dat vervolgens te verkopen aan adverteerders; daarnaast blijken een aantal instituten voor nationale veiligheid zoveel mogelijk informatie van het internet af te tappen als ze aankunnen.

In sommige gevallen is een dergelijke controle van gedrag en activiteiten – dat wil zeggen, *surveillance* – wenselijk. Belastingdiensten moeten bijvoorbeeld op de hoogte zijn van het inkomen en spaargeld van mensen om hun werk te kunnen doen, en veel mensen delen bewust regelmatig allerlei gegevens met de rest van de wereld op Facebook. Er kan echter ook sprake zijn van ernstige juridische of morele problemen, bijvoorbeeld wanneer een repressieve overheid surveillance gebruikt om kritiek te onderdrukken. Mensen hebben ook niet altijd de mogelijkheid om iets aan surveillance te doen, ook al zijn ze het er niet mee eens: bijvoorbeeld, als iemand die geen burger is van de Verenigde Staten heb ik geen enkele juridische of democratische manier om te voorkomen of invloed uit te oefenen op de informatie die de NSA over me verzamelt door het analyseren van internetverkeer. Verder is men zich er vaak niet eens van bewust; de omvang van de surveillance van de NSA en vergelijkbare instituten was grotendeels onbekend voor de onthullingen van Edward Snowden hierover in 2013, en de meeste mensen hebben nauwelijks door hoeveel informatie ze over zichzelf prijsgeven wanneer ze gebruik maken van het internet. Ook gebeurt het dat instituten of bedrijven veel meer data verzamelen dan ze nodig hebben voor hun doelen. Samenvattend is het duidelijk dat surveillance op vele manieren een bedreiging vormt voor de privacy van mensen, en daarnaast is het in ieder geval vanwege de volgende drie redenen gevaarlijk:

- Als alles wat je doet altijd wordt opgeslagen door grote bedrijven en overheidsinstellingen, dan zouden deze instellingen deze informatie later tegen je kunnen gebruiken. Dit soort instellingen zeggen vaak dat ze te vertrouwen zijn met deze informatie, maar zelfs als dat zo is dan blijft de vraag of ze dat in de toekomst nog steeds zijn. Misschien leven we over vijftig jaar in een totalitaire staat; als zo'n staat alles van iedereen weet zou er niet aan te ontkomen zijn.
- Zodra data over wat je doet en wie je bent buiten je controle wordt opgeslagen, is het vaak zeer moeilijk om bij te houden waar en hoe het wordt opgeslagen, wat ermee gedaan wordt, en wie er toegang toe heeft. Bijvoorbeeld, *privacy policy*'s die

aangeven wat gedaan wordt met je data worden vaak zonder bericht gewijzigd; de overheid kan later (geheime) wetten of regelgeving invoeren die haar toegang tot de data verschaft; en de data kan worden gelekt of gestolen. Aangezien dit soort data tegenwoordig bijna eindeloos kan worden opgeslaan is het risico dat dit soort dingen gebeurt groot, en alledrie de genoemde voorbeelden zijn intussen ook al een aantal maal voorgekomen. Dit probleem vergroot ook de ernst van het eerste punt hierboven.

- Wanneer mensen weten dat ze in de gaten gehouden worden doen ze misschien bepaalde dingen niet die ze anders wel gedaan zouden hebben. Angst voor mogelijke consequenties leidt dan tot onderdrukking van kritische geluiden en handelingen, resulterend in zelfcensuur. Dit staat volkomen haaks op het ideaalbeeld van een open en democratische maatschappij.

In het kort: in een maatschappij waarin informatie hetzelfde is als geld en macht is het onverstandig om je persoonlijke informatie weg te geven. Het moet gezien en behandeld worden als een soort grondstof. Daarom wordt het steeds belangrijker dat mensen zich bewust zijn van en in controle zijn van wie wat over ze weet. Er is, met andere woorden, een noodzaak voor *identiteitsbeheer*: het verkrijgen, gebruiken en beschermen van persoonlijke informatie, online of offline, vaak voor het verkrijgen van toegang tot services. We zijn echter niet op zoek naar manieren om alle persoonlijke informatie constant verborgen te houden, omdat dit simpelweg niet mogelijk zou zijn: bijvoorbeeld, een klant in de slijterij zal de kassamedewerker er op een of andere manier van moeten overtuigen dat zij ouder is dan 18. Echter, de kassamedewerker hoeft niet meer over haar te weten dan enkel dat feit om een alcoholische drank aan haar te kunnen verkopen.

Feiten zoals dat de klant ouder is dan 18, of bijvoorbeeld dat ze een abonnement heeft op een krant, noemen we *attributen*. Een paspoort kan dan gezien worden als een lijst van zulke attributen die gemaakt is door de overheid op een onvervalsbare manier. Het voorbeeld hierboven laat zien dat we in staat willen zijn om zulke attributen selectief te laten zien aan anderen, zodat de klant in de slijterij kan bewijzen dat ze inderdaad ouder is dan 18, zonder haar precieze leeftijd, naam of andere details prijs te geven die niet relevant zijn voor de transactie.

In het tweede deel van dit proefschrift proberen we deze doelen te bereiken door middel van cryptografische middelen genaamd *attribuut-gebaseerde credentialschema's*. In zulke schema's krijgen gebruikers een credential van een uitgever (Engels: issuer). Ieder credential kan meerdere attributen bevatten (die een beperkte hoeveelheid informatie bevatten), samen met een digitale handtekening gemaakt door de uitgever over deze attributen. De gebruiker kan dan aan anderen selectief een aantal van deze attributen laten zien, terwijl de andere attributen verborgen blijven. De ontvangende partij (genaamd de controleur, Engels: verifier) kan de handtekening van de uitgever gebruiken om zeker te weten dat het credential inderdaad gemaakt is door de uitgever. Als de overheid bijvoorbeeld dergelijke attribuut-gebaseerde credentials zou uitgeven aan haar burgers, zou ik ervoor kunnen kiezen om al mijn persoonsgegevens vrij te geven bij de douane in het vliegveld, of ik zou kunnen bewijzen dat ik ouder ben dan 18 in een winkel.

Sommige credentialschema's hebben daarnaast een interessante eigenschap genaamd *multi-show onlinkbaarheid*: wanneer een controleur twee credentials te zien krijgt kan hij onmogelijk weten of hij twee keer hetzelfde credential zag, of twee verschillende (tenminste, zolang de vrijgegeven attributen de credentials niet van elkaar onderscheiden). Een credentialschema kan bovendien tegelijkertijd of apart daarvan *uitgever-onlinkbaar* zijn, wat betekent dat de uitgever het gebruik van een credential onmogelijk kan koppelen aan een uitgifte van een credential.

We zullen ons met name bezig houden met de volgende eigenschappen die credentialschema's kunnen hebben.

Veiligheid. Om ervoor te zorgen dat de controleur er echt van overtuigd kan zijn dat de gebruiker een geldig credential heeft (in het bijzonder dat hij dit credential gekregen heeft van de uitgever), moet het bewijsbaar zijn dat gebruikers credentials niet zonder kennis en toestemming van de uitgever kunnen maken. Ook wanneer een schema bepaalde anonimiteitseigenschappen heeft moet dit bewijsbaar zijn.

Efficiëntie. Een credentialschema moet praktisch zijn in gebruik. Een manier om dit te bereiken zou kunnen zijn het implementeren ervan op smart cards, die zeer beperkte capaciteiten hebben. Daarom is het belangrijk dat schema's zo efficiënt mogelijk zijn.

Attribuut-gebaseerd. Zoals hierboven beschreven wil men in verschillende situaties verschillende attributen kunnen vrijgeven. We zullen echter ook een aantal credentialschema's bestuderen die niet attribuut-gebaseerd zijn, of die zelfs helemaal geen informatie op kunnen slaan.

Anonimiteit. Waar mogelijk moet een credentialschema zoveel mogelijk gegevens van de gebruiker verbergen, of zoveel als de gebruiker wenst. Hoewel dit al gedeeltelijk bereikt kan worden door irrelevante attributen verborgen te houden, zullen we vooral geïnteresseerd zijn in schema's die één maar liefst beide vormen van onlinkbaarheid bieden.

We behandelen vooral schema's die gebruik maken van bilineaire groep-paren: een paar van twee priem-orde elliptische krommen dat een speciale afbeelding genaamd de *paring* toelaat. Elliptische krommen bieden hoge veiligheid tegelijkertijd met kleine groepselementen en efficiënte algoritmes voor de groepsoperaties, en met de paring kunnen de relaties tussen specifieke groepselementen bestudeerd worden. Het tweede deel van dit proefschrift gaat dan ook over de volgende vraag: *Bestaat er, of kunnen we een credentialschema maken die gebruik maakt van bilineaire groep-paren en die alle bovengenoemde doelen haalt, inclusief onlinkbaarheid? Tot in hoeverre moet er een afweging gemaakt worden tussen deze doelen?*

De credentialschema's die we bestuderen zullen steeds meer van de bovenstaande doelen tegelijk halen. In hoofdstukken 6 en 8 bestuderen we een aantal schema's die efficiënt, veilig en onlinkbaar tegelijkertijd hadden moeten zijn, maar daar niet in slaagden. Daarnaast zullen we inderdaad een afweging vinden tussen anonimiteit en efficiëntie, in de schema's van hoofdstukken 7 en 9; het ene hoofdstuk behandelt een zeer efficiënt schema die geen onlinkbaarheid biedt, terwijl het schema van het andere hoofdstuk weliswaar minder efficiënt is, maar toch het eerste schema is dat allevier doelen tegelijkertijd behaalt.

Samenvatting

Het tweede deel van dit proefschrift bevat de volgende hoofdstukken.

Hoofdstuk 5 behandelt enkele notaties en fundamentele concepten die in het hele tweede deel van dit proefschrift van pas komen, waaronder: wanneer iets efficiënt berekenbaar is; cyclische groepen en bilineaire paringen, *zero-knowledge* bewijzen en digitale handtekeningschema's. Hier definiëren we ook credentialschema's, het hoofdonderwerp van de latere hoofdstukken.

Hoofdstuk 6 bestudeert een aantal (niet-attribuut-gebaseerde) credentialschema's uit de literatuur, die kapot zijn in de zin dat ze niet de eigenschappen blijken te hebben die ze hadden moeten hebben. We bewijzen een stelling die zegt dat het in bepaalde omstandigheden moeilijk is om onlinkbaarheid en onvervalsbaarheid te verenigen in zulke schema's. Dit hoofdstuk is gebaseerd op het volgende artikel.

> [HLR15] J.-H. Hoepman, W. Lueks en S. Ringers. "On Linkability and Malleability in Self-blindable Credentials". In: *Information Security Theory and Practice: 9th IFIP WG 11.2 International Conference, WISTP 2015*. Red. door N. R. Akram en S. Jajodia. Cham: Springer International Publishing, 2015, p. 203–218.

Hoofdstuk 7 introduceert een interactief algoritme voor het tekenen van een generalisatie van Boneh–Boyen handtekeningen, die een gedeelte van het bericht tegelijkertijd met de resulterende handtekening verbergt voor de handtekening-uitgever. Door dit toe te passen vinden we een attribuut-gebaseerd credentialschema, die eenvoudig en zeer efficiënt is maar die geen multi-show onlinkbaarheid biedt (hoewel het wel uitgever-onlinkbaar is). Vergeleken met zijn bekendste concurrent, U-Prove, biedt ons schema vergelijkbare efficiëntie maar sterkere veiligheidsgaranties.

Hoofdstuk 8 heeft het attribuut-gebaseerde credentialschema van Hanzlik en Kluczniak [HK14] als onderwerp. Dit schema had een multi-show onlinkbare variant van U-Prove moeten zijn (die, net als ons schema van hoofdstuk 7, niet multi-show onlinkbaar maar zeer efficiënt is). We tonen een aanval op het schema van Hanzlik en Kluczniak, waarmee een aantal gebruikers samen een nieuw credential kan maken met daarin attributen van hun keuze, zonder medewerking of kennis van de uitgever. Daarnaast wijzen we de fout in hun paper aan. Dit hoofdstuk is gebaseerd op het volgende artikel.

> [VRH16] E. Verheul, S. Ringers en J.-H. Hoepman. "The self-blindable U-Prove scheme from FC'14 is forgeable". In: *Financial Cryptography and Data Security – FC'16* (2016). In print. URL: https://eprint.iacr.org/2015/725.

Hoofdstuk 9 introduceert tenslotte een tweede attribuut-gebaseerd credentialschema die wel multi-show onlinkbaar is, en die alle vier doelen uit de vorige sectie haalt. Zijn veiligheid volgt uit ons bewijs dat hij onvervalsbaar is, en we laten ook zien dat het schema onlinkbaar is, zodat de anonimiteit van de gebruiker goed beschermd wordt. Hoewel dit schema niet zo efficiënt is als dat van hoofdstuk 7, is hij wel

efficiënter dan alle vergelijkbare (in het bijzonder onlinkbare) schema's die ons bekend zijn. Aan het eind van het hoofdstuk vergelijken we ons schema kort met die van de eerdere hoofdstukken.

Acknowledgments

Try as I might, I can think of no way to convey the depth of my gratitude to my promotor, Jaap Top, for everything that he has done for me and this project, and for the enormous amount of faith he always had in me. He has supported and helped me in ways that I had no reason to expect from him, and even if he had not, he would still have been an excellent promotor. At times when I no longer saw solutions, he always saw ways to go forward.

Secondly, I am very grateful to Arthemy Kiselev, my supervisor for the first part of this thesis. He not only introduced me to the beautiful subjects of jet space geometry and quantization, but I also learned much from him about, for example, writing scientific articles and conducting oneself in the scientific community. Apart from this I have fond memories of the many stories and anecdotes he told me, and of our animated discussions in his dimly lit office.

I am equally grateful to my supervisor for the second part of this thesis: Jaap-Henk Hoepman, for taking me in and having trust in me. Computer science, security and privacy aspects have always been of great interest to me, and I am very happy to have been given the chance to pursue these interests. His encouragement, his critical attitude towards my work, and his focus on properly presenting your work has been very important and helpful to me.

I am also very grateful to Bart Jacobs, for arranging the extension that allowed me to work on the IRMA project, for introducing me to the IRMA project itself as well as everything surrounding it, and for his advice and guidance in the last year. The extension gave me some much-needed time to properly finish the second part of this project, and I have very much enjoyed contributing to the IRMA project, of which the rather practical nature was an excellent complement to the theoretical work of this thesis.

Thanks also to Klaas Landsman, for his advice at an important time, and his continued interest in me and this project throughout its duration.

Next I wish to thank both the universities of Groningen and Nijmegen for the warm atmosphere they provided, allowing me to do this work; and my colleagues at both

universities, for their companionship and the many enjoyable lunches we had. Particular thanks goes to the university of Groningen for giving me the opportunity to teach exercise classes, as well as to design a course and lecture it. Some of the best moments of these years were when I was teaching, and I am very happy to have learned how immensely satisfying teaching can be.

Particular thanks go to the co-authors of my published papers: Arthemy Kiselev, Jaap-Henk Hoepman, Wouter Lueks and Eric Verheul, for our constructive collaborations, for those parts of our papers that they wrote, and for their many attentive suggestions and corrections for the parts that I wrote. Fabian van den Broek also deserves mentioning here; although he and I never published a paper together, we have enjoyed many months of fruitful collaboration (together with Wouter) on the IRMA project.

Last but certainly not least, I am immensely thankful to my partner in life, Tomas Harreveld, for his continuous support and encouragement through the highs and lows of these five years; for sharing my interest in the second subject of this thesis; for all the advice he gave me; for the faith that he always had in me, even at times when I saw no reason for it; and for listening with great attention and interest to my endless stories about both subjects of this thesis, and everything that transpired as it came about.

Vleuten, August 2016

Biography

Sietse Ringers was born in 1984. After having obtained his bachelor's degree in Physics and Astronomy, he graduated in Mathematical Physics at the University of Amsterdam in 2011, under supervision of Robbert Dijkgraaf. Since then he has been a PhD student at the University of Groningen, studying the connections between differential geometry and theoretical physics under supervision of Arthemy Kiselev, and identity management and credential schemes under supervision of Jaap-Henk Hoepman.

In his spare time he enjoys programming (in particular on the IRMA project which implements many of the techniques from the second part of this thesis), listening to (classical) music and singing in choirs. He and his partner currently live in Vleuten in The Netherlands.

Bibliography

Quantization using Jet Space Geometry

[Bay+77] F. Bayen, M. Flato, C. Fronsdal, A. Lichnerowicz, and D. Sternheimer. "Quantum mechanics as a deformation of classical mechanics". In: *Letters in Mathematical Physics* 1.6 (1977), pp. 521–530 (cit. on pp. xii, 45).

[Bay+78a] F. Bayen, M. Flato, C. Fronsdal, A. Lichnerowicz, and D. Sternheimer. "Deformation theory and quantization. I. Deformations of symplectic structures". In: *Annals of Physics* 111.1 (1978), pp. 61–110 (cit. on pp. xii, 45).

[Bay+78b] F. Bayen, M. Flato, C. Fronsdal, A. Lichnerowicz, and D. Sternheimer. "Deformation theory and quantization. II. Physical applications". In: *Annals of Physics* 111.1 (1978), pp. 111–151 (cit. on pp. xii, 45).

[BRS75] C. Becchi, A. Rouet, and R. Stora. "Renormalization of the abelian Higgs-Kibble model". In: *Commun. Math. Phys.* 42 (June 1975), pp. 127–162 (cit. on pp. xi, 11, 25).

[BRS76] C. Becchi, A. Rouet, and R. Stora. "Renormalization of gauge theories". In: *Ann. Phys.* 98 (June 1976), pp. 287–321 (cit. on pp. xi, 11, 25).

[BV81] I. A. Batalin and G. A. Vilkovisky. "Gauge algebra and quantization". In: *Phys. Lett. B* 102 (June 1981), pp. 27–31 (cit. on pp. xii, 11, 25).

[BV83] I. A. Batalin and G. A. Vilkovisky. "Quantization of gauge theories with linearly dependent generators". In: *Phys. Rev. D* 28 (10 Nov. 1983), pp. 2567–2582 (cit. on pp. xii, 11, 25).

[CF00] A. S. Cattaneo and G. Felder. "A path integral approach to Kontsevich quantization formula". In: *Commun. Math. Phys.* 2012 (3 2000), pp. 591–611. arXiv: `math/9902090` (cit. on pp. 11, 26, 39, 41).

[CI] A. S. Cattaneo and D. Indelicato. "Formality and Star Products". Lecture notes of a course at PQR2003 Euroschool, 2003 (cit. on pp. 50, 57).

[Dit99] G. Dito. "Kontsevich Star Product on the Dual of a Lie Algebra". In: *Letters in Mathematical Physics* 48.4 (1999), pp. 307–322. arXiv: math/9905080 (cit. on p. 67).

[Dor93] I. Dorfman. *Dirac structures and integrability of nonlinear evolution equations*. Nonlinear science. Wiley & Sons, 1993 (cit. on p. 11).

[Fed94] B. V. Fedosov. "A simple geometrical construction of deformation quantization". In: *J. Differential Geom.* 40.2 (1994), pp. 213–238 (cit. on p. xii).

[GD80] I. M. Gel'fand and I. Dorfman. "The Schouten Bracket and Hamiltonian operators". In: *Functional Analysis and Its Applications* 14.3 (1980), pp. 223–226 (cit. on p. 11).

[GR99] S. Gutt and J. Rawnsley. "Equivalence of star products on a symplectic manifold; an introduction to Deligne's Čech cohomology classes". In: *Journal of Geometry and Physics* 29.4 (1999), pp. 347–392 (cit. on p. 54).

[Gut83] S. Gutt. "An explicit *-product on the cotangent bundle of a Lie group". In: *Letters in Mathematical Physics* 7.3 (1983), pp. 249–258 (cit. on p. 67).

[Ham80] R. W. Hamming. "The unreasonable effectiveness of mathematics". In: *Amer. Math. Monthly* 87.2 (Feb. 1980). URL: https://www.dartmouth.edu/~matc/MathDrama/reading/Hamming.html (cit. on p. xi).

[HT92] M. Henneaux and C. Teitelboim. *Quantization of gauge systems*. Princeton, NJ: Princeton University Press, 1992, pp. xxviii+520 (cit. on pp. 11, 25, 31).

[Kat00] V. Kathotia. "Kontsevich's universal formula for deformation quantization and the Campbell–Baker–Hausdorff formula". In: *International Journal of Mathematics* 11.4 (2000), pp. 523–551. arXiv: math/9811174 (cit. on p. 67).

[Kis12a] A. V. Kiselev. "Homological evolutionary vector fields in Korteweg-de Vries, Liouville, Maxwell, and several other models". In: *J. Phys. Conf. Ser.* 343.1 (Feb. 2012), p. 012058. arXiv: 1111.3272 (cit. on p. 25).

[Kis12b] A. V. Kiselev. "On the variational noncommutative Poisson geometry". In: *Physics of Particles and Nuclei* 43 (2012), pp. 663–665. arXiv: 1112.5784 (cit. on p. 11).

[Kis12c] A. V. Kiselev. "The twelve lectures in the (non)commutative geometry of differential equations". 140p, preprint IHÉS M-12-13. 2012. URL: http://preprints.ihes.fr/2012/M/M-12-13.pdf (cit. on pp. xi, 3, 12, 24, 25, 38, 71, 72).

[KKV04] P. Kersten, I. Krasil'shchik, and A. Verbovetsky. "Hamiltonian operators and l^*-coverings". In: *J. Geom. Phys.* 50.1-4 (2004), pp. 273–302. arXiv: math/0304245 (cit. on pp. 8, 11).

[Kon03] M. Kontsevich. "Deformation Quantization of Poisson Manifolds". In: *Letters in Mathematical Physics* 66.3 (2003), pp. 157–216. arXiv: q-alg/9709040 (cit. on pp. xii, 46, 50, 52, 53, 58, 61, 66).

[Kon93] M. Kontsevich. "Formal (Non)-Commutative Symplectic Geometry". In: *The Gelfand Mathematical Seminars, 1990–1992*. Ed. by I. Gelfand, L. Corwin, and J. Lepowsky. Birkhäuser Boston, 1993, pp. 173–187 (cit. on pp. 11, 24).

[KR12] A. V. Kiselev and S. Ringers. "A comparison of definitions for the Schouten bracket on jet spaces". In: *Proceedings of the Sixth International Workshop "Group Analysis of Differential Equations and Integrable Systems"*. Larnaca, Cyprus, 2012, 15p. arXiv: 1208.6196 (cit. on pp. x, 3, 189).

[KV11] I. S. Krasil'shchik and A. Verbovetsky. "Geometry of jet spaces and integrable systems". In: *J. Geom. Phys.* 61.9 (2011), pp. 1633–1674. arXiv: 1002.0077 (cit. on pp. xi, 3, 11, 12, 19, 21, 26).

[KV99] I. S. Krasil'shchik and A. M. Vinogradov, eds. *Symmetries and conservation laws for differential equations of mathematical physics*. Vol. 182. Translations of Mathematical Monographs. Providence, RI: Amer. Math. Soc., 1999, pp. xiv+333 (cit. on pp. xi, 3, 26, 71).

[Lic77] A. Lichnerowicz. "Les variétés de Poisson et leurs algèbres de Lie associées". In: *Journal of Differential Geometry* 12.2 (1977), pp. 253–300 (cit. on p. 11).

[Lic78] A. Lichnerowicz. "Les variétés de Poisson et leurs algèbres de Lie associées". In: *J. Math. Pures Appl.* 57 (1978), pp. 453–488 (cit. on p. 11).

[Nij55] A. Nijenhuis. "Jacobi-type identities for bilinear differential concomitants of certain tensor fields I". In: *Indag. Math.* 17 (1955), pp. 390–403 (cit. on p. 11).

[Olv00] P. J. Olver. *Applications of Lie groups to differential equations*. Second Edition. Vol. 107. Graduate Texts in Mathematics. New York: Springer-Verlag, 2000, p. 541 (cit. on pp. xi, 3).

[OS98] P. J. Olver and V. V. Sokolov. "Integrable Evolution Equations on Associative Algebras". In: *Communications in Mathematical Physics* 193.2 (1998), pp. 245–268 (cit. on pp. 11, 24).

[RS03] A. Reyman and M. A. Semenov–Tyan-Shansky. "Integrable systems: group-theoretic approach". In: *Reg. Chaot. Dyn.* (2003). in Russian (cit. on p. 11).

[RS94] A. G. Reyman and M. A. Semenov–Tian-Shansky. "Group-Theoretical Methods in the Theory of Finite-Dimensional Integrable Systems". In: *Dynamical Systems VII*. Ed. by V. I. Arnol'd and S. P. Novikov. Vol. 16. Encyclopaedia of Mathematical Sciences. Springer Berlin Heidelberg, 1994, pp. 116–225 (cit. on p. 11).

[Sch40] J. A. Schouten. "Über Differentialkonkomitanten zweier kontravarianten Grössen". In: *Indag. Math.* 2 (1940), pp. 449–452 (cit. on p. 11).

[Sch53] J. A. Schouten. "On the differential operators of the first order in tensor calculus". In: *Convegno Int. Geom. Diff. Italia.* (1953), pp. 1–7 (cit. on p. 11).

[Sch93] A. Schwarz. "Geometry of Batalin-Vilkovisky quantization". In: *Commun. Math. Phys.* 155 (July 1993), pp. 249–260. arXiv: hep-th/9205088 (cit. on pp. 26, 29).

[Sch94] A. Schwarz. "Symmetry transformations in Batalin-Vilkovisky formalism". In: *Letters in Mathematical Physics* 31.4 (1994), pp. 299–301 (cit. on pp. 26, 29).

[Sch99] A. Schwarz. "Quantum Observables, Lie Algebra Homology and TQFT". In: *Letters in Mathematical Physics* 49.2 (1999), pp. 115–122. arXiv: hep-th/9904168 (cit. on pp. 26, 29).

[Tyu75] I. V. Tyutin. "Gauge Invariance in Field Theory and Statistical Physics in Operator Formalism". Lebedev Physics Institute preprint 39. 1975 (cit. on pp. xi, 11, 25).

[Vin01] A. M. Vinogradov. *Cohomological Analysis of Partial Differential Equations and Secondary Calculus*. Vol. 204. Translations of Mathematical Monographs. Amer. Math. Soc., 2001 (cit. on pp. xi, 3, 26).

[Vit09] L. Vitagliano. "Secondary calculus and the covariant phase space". In: *Journal of Geometry and Physics* 59.4 (2009), pp. 426–447. arXiv: 0809.4164 (cit. on p. 26).

[Vor02] T. Voronov. "Graded manifolds and Drinfeld doubles for Lie bialgebroids". In: *Quantization, Poisson brackets and beyond (Manchester, 2001)*. Vol. 315. Contemp. Math. Providence, RI: Amer. Math. Soc., 2002, pp. 131–168. arXiv: math/0105237 (cit. on p. 14).

[Wig60] E. P. Wigner. "The unreasonable effectiveness of mathematics in the natural sciences". In: *Comm. Pure Appl. Math.* 13 (Feb. 1960). URL: https://www.dartmouth.edu/~matc/MathDrama/reading/Wigner.html (cit. on p. xi).

[Wit90] E. Witten. "A note on the antibracket formalism". In: *Modern Phys. Lett. A* 5.7 (1990), pp. 487–494 (cit. on pp. 11, 25).

[Zin75] J. Zinn-Justin. "Renormalization of gauge theories". In: *Trends in elementary particle theory*. Vol. 37. Lect. notes in phys. Berlin: Springer, 1975, pp. 1–39 (cit. on p. 11).

Identity Management using Credential Schemes

[ACM05] G. Ateniese, J. Camenisch, and B. de Medeiros. "Untraceable RFID Tags via Insubvertible Encryption". In: *Proceedings of the 12th ACM Conference on Computer and Communications Security - CCS '05*. New York, NY, USA: ACM, 2005, pp. 92–101. URL: http://doi.acm.org/10.1145/1102120.1102134 (cit. on pp. 153, 154).

[Adj+15] G. Adj, A. Menezes, T. Oliveira, and F. Rodríguez-Henríquez. "Computing Discrete Logarithms in $\mathbb{F}_{3^{6\cdot 137}}$ and $\mathbb{F}_{3^{6\cdot 163}}$ Using Magma". In: *Arithmetic of Finite Fields*. Ed. by Ç. K. Koç, S. Mesnager, and E. Savaş. Vol. 9061. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 3–22 (cit. on p. 92).

[AF07] M. Abe and S. Fehr. "Perfect NIZK with Adaptive Soundness". In: *Theory of Cryptography*. Ed. by S. P. Vadhan. Vol. 4392. LNCS. Springer Berlin Heidelberg, 2007, pp. 118–136 (cit. on p. 175).

[AF96] M. Abe and E. Fujisaki. "How to date blind signatures". In: *Advances in Cryptology — ASIACRYPT '96*. Ed. by K. Kim and T. Matsumoto. Vol. 1163. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 244–251 (cit. on pp. 125, 141).

[AHS11] G. Alpár, J. Hoepman, and J. Siljee. "The Identity Crisis. Security, Privacy and Usability Issues in Identity Management". In: *CoRR* abs/1101.0427 (2011). URL: http://arxiv.org/abs/1101.0427 (cit. on p. xvii).

[Alp15] G. Alpár. "Attribute-Based Identity Management: Bridging the Cryptographic Design of ABCs with the Real World". PhD thesis. Radboud University, Nijmegen, The Netherlands, 2015 (cit. on p. xvii).

[Ant+02] A. Antipa, D. Brown, A. Menezes, R. Struik, and S. Vanstone. "Validation of Elliptic Curve Public Keys". In: *Public Key Cryptography – PKC 2003*. Ed. by Y. G. Desmedt. Vol. 2567. LNCS. Springer Berlin Heidelberg, 2002, pp. 211–223 (cit. on p. 161).

[AO00] M. Abe and T. Okamoto. "Provably Secure Partially Blind Signatures". In: *Advances in Cryptology – CRYPTO 2000*. Ed. by M. Bellare. Vol. 1880. LNCS. Springer Berlin Heidelberg, 2000, pp. 271–286 (cit. on pp. 126, 127, 141).

[Bar01] B. Barak. "How to Go Beyond the Black-Box Simulation Barrier". In: *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. IEEE Computer Society, 2001, pp. 106–115 (cit. on p. 97).

[BB08] D. Boneh and X. Boyen. "Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups". In: *J. Cryptology* 21.2 (2008), pp. 149–177 (cit. on pp. 119, 126, 130, 131, 156).

[Bel+02] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. "The Power of RSA Inversion Oracles and the Security of Chaum's RSA-Based Blind Signature Scheme". In: *Financial Cryptography: 5th International Conference, FC 2001*. Ed. by P. Syverson. Vol. 2339. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 319–338 (cit. on p. 141).

[Bel+09] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. "Randomizable Proofs and Delegatable Anonymous Credentials". In: *Advances in Cryptology - CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*. Ed. by S. Halevi. Vol. 5677. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 108–125 (cit. on p. 141).

[Beu+10] J. Beuchat, J. E. González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya. "High-Speed Software Implementation of the Optimal Ate Pairing over Barreto-Naehrig Curves". In: *Pairing-Based Cryptography - Pairing 2010*. Ed. by M. Joye, A. Miyaji, and A. Otsuka. Vol. 6487. Lecture Notes in Computer Science. Springer, 2010, pp. 21–39 (cit. on pp. 170, 171).

[BF01] D. Boneh and M. K. Franklin. "Identity-Based Encryption from the Weil Pairing". In: *Advances in Cryptology - CRYPTO 2001*. Ed. by J. Kilian. Vol. 2139. LNCS. Springer, 2001, pp. 213–229 (cit. on p. 156).

[Bha+09] R. Bhaskar, K. Chandrasekaran, S. V. Lokam, P. L. Montgomery, R. Venkatesan, and Y. Yacobi. "An Observation about Variations of the Diffie-Hellman Assumption". In: *Serdica Journal of Computing* 3 (2009) (cit. on pp. 131, 139).

[Bit+12] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer. "From Extractable Collision Resistance to Succinct Non-interactive Arguments of Knowledge, and Back Again". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference - ITCS '12*. Cambridge, Massachusetts: ACM, 2012, pp. 326–349 (cit. on p. 174).

[Bit+14] N. Bitansky, R. Canetti, A. Chiesa, S. Goldwasser, H. Lin, A. Rubinstein, and E. Tromer. "The Hunting of the SNARK". In: *IACR Cryptology ePrint Archive* 2014 (2014). URL: https://eprint.iacr.org/2014/580 (cit. on p. 174).

[BL13a] F. Baldimtsi and A. Lysyanskaya. "Anonymous Credentials Light". In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. CCS '13. Berlin, Germany: ACM, 2013, pp. 1087–1098 (cit. on pp. 141, 153).

[BL13b] F. Baldimtsi and A. Lysyanskaya. "On the Security of One-Witness Blind Signature Schemes". In: *Advances in Cryptology - ASIACRYPT 2013*. Ed. by K. Sako and P. Sarkar. Vol. 8270. LNCS. Springer Berlin Heidelberg, 2013, pp. 82–99 (cit. on pp. 126, 139, 152, 153).

[Bla+13] O. Blazy, G. Fuchsbauer, D. Pointcheval, and D. Vergnaud. "Short blind signatures". In: *Journal of Computer Security* 21.5 (2013), pp. 627–661 (cit. on p. 141).

[BLS04] D. Boneh, B. Lynn, and H. Shacham. "Short Signatures from the Weil Pairing". In: *J. Cryptology* 17.4 (2004), pp. 297–319 (cit. on pp. 103, 156).

[BN06] P. S. L. M. Barreto and M. Naehrig. "Pairing-Friendly Elliptic Curves of Prime Order". In: *Selected Areas in Cryptography*. Ed. by B. Preneel and S. Tavares. Vol. 3897. LNCS. Springer Berlin Heidelberg, 2006, pp. 319–331 (cit. on p. 93).

[Bol02] A. Boldyreva. "Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme". In: *Public Key Cryptography — PKC 2003*. Ed. by Y. G. Desmedt. Vol. 2567. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 31–46 (cit. on p. 141).

[BP04a] M. Bellare and A. Palacio. "The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols". English. In: *Advances in Cryptology – CRYPTO 2004*. Ed. by M. Franklin. Vol. 3152. LNCS. Springer Berlin Heidelberg, 2004, pp. 273–289 (cit. on p. 174).

[BP04b] M. Bellare and A. Palacio. "Towards Plaintext-Aware Public-Key Encryption Without Random Oracles". English. In: *Advances in Cryptology - ASIACRYPT 2004*. Ed. by P. Lee. Vol. 3329. LNCS. Springer Berlin Heidelberg, 2004, pp. 48–62 (cit. on p. 174).

[BPW12] D. Bernhard, O. Pereira, and B. Warinschi. "How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios". In: *Advances in Cryptology – ASIACRYPT 2012*. Ed. by X. Wang and K. Sako. Vol. 7658. LNCS. Springer Berlin Heidelberg, 2012, pp. 626–643 (cit. on p. 169).

[BR93] M. Bellare and P. Rogaway. "Random oracles are practical: a paradigm for designing efficient protocols". In: *ACM Conference on Computer and Communications Security*. 1993, pp. 62–73 (cit. on p. 169).

[Bra00] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000 (cit. on pp. xvii, 88, 89, 98, 125, 126, 151, 153).

[BSS05] I. F. Blake, G. Seroussi, and N. P. Smart, eds. *Advances in Elliptic Curve Cryptography*. Cambridge Books Online. Cambridge University Press, 2005 (cit. on p. 93).

[CDM00] R. Cramer, I. Damgård, and P. MacKenzie. "Efficient Zero-Knowledge Proofs of Knowledge without Intractability Assumptions". In: *Public Key Cryptography*. Ed. by H. Imai and Y. Zheng. Vol. 1751. LNCS. Springer Berlin Heidelberg, 2000, pp. 354–372 (cit. on pp. 96, 99, 130).

[CDN01] R. Cramer, I. Damgård, and J. B. Nielsen. "Multiparty Computation from Threshold Homomorphic Encryption". In: *Advances in Cryptology — EUROCRYPT 2001*. Ed. by B. Pfitzmann. Vol. 2045. LNCS. Extended version at `http://www.brics.dk/RS/00/14/BRICS-RS-00-14.pdf`. Springer Berlin Heidelberg, 2001, pp. 280–300 (cit. on p. 130).

[CDS94] R. Cramer, I. Damgård, and B. Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols". In: *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology - CRYPTO '94*. London, UK, UK: Springer-Verlag, 1994, pp. 174–187 (cit. on p. 99).

[CFN90] D. Chaum, A. Fiat, and M. Naor. "Untraceable Electronic Cash". English. In: *Advances in Cryptology — CRYPTO' 88*. Ed. by S. Goldwasser. Vol. 403. LNCS. Springer New York, 1990, pp. 319–327 (cit. on p. 125).

[CGH04] R. Canetti, O. Goldreich, and S. Halevi. "The Random Oracle Methodology, Revisited". In: *J. ACM* 51.4 (2004), pp. 557–594 (cit. on p. 169).

[Cha83] D. Chaum. "Blind Signatures for Untraceable Payments". In: *Advances in Cryptology*. Ed. by D. Chaum, R. L. Rivest, and A. T. Sherman. Springer US, 1983, pp. 199–203 (cit. on pp. 125, 126, 140).

[Cha90] D. Chaum. "Showing credentials without identification transferring signatures between unconditionally unlinkable pseudonyms". In: *Advances in Cryptology — AUSCRYPT '90*. Ed. by J. Seberry and J. Pieprzyk. Vol. 453. LNCS. Springer Berlin Heidelberg, 1990, pp. 245–264 (cit. on p. 125).

[Cho+05] S. S. M. Chow, L. C. K. Hui, S. M. Yiu, and K. P. Chow. "Two Improved Partially Blind Signature Schemes from Bilinear Pairings". In: *Information Security and Privacy: 10th Australasian Conference, ACISP 2005*. Ed. by C. Boyd and J. M. González Nieto. Vol. 3574. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 316–328 (cit. on p. 141).

[CHP07] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen. "Batch Verification of Short Signatures". In: *Advances in Cryptology - EUROCRYPT 2007*. Ed. by M. Naor. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 246–263. URL: https://eprint.iacr.org/2007/172.pdf (cit. on pp. 153, 154).

[CJFE] *Snowden Surveillance Archive*. Canadian Journalists for Free Expression. URL: https://cjfe.org/snowden (cit. on p. xvii).

[CKW05] J. Camenisch, M. Koprowski, and B. Warinschi. "Efficient Blind Signatures Without Random Oracles". In: *Security in Communication Networks: 4th International Conference, SCN 2004*. Ed. by C. Blundo and S. Cimato. Vol. 3352. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 134–148 (cit. on p. 141).

[CL01] J. Camenisch and A. Lysyanskaya. "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation". In: *Advances in Cryptology — EUROCRYPT 2001*. Ed. by B. Pfitzmann. Vol. 2045. LNCS. Springer Berlin Heidelberg, 2001, pp. 93–118 (cit. on pp. xvii, 87, 112, 123, 151, 153).

[CL02] J. Camenisch and A. Lysyanskaya. "A Signature Scheme with Efficient Protocols". In: *Security in Communication Networks, Third International Conference, SCN 2002*. Ed. by S. Cimato, C. Galdi, and G. Persiano. Vol. 2576. LNCS. Springer, 2002, pp. 268–289 (cit. on pp. 123, 159).

[CL04] J. Camenisch and A. Lysyanskaya. "Signature Schemes and Anonymous Credentials from Bilinear Maps". English. In: *Advances in Cryptology – CRYPTO 2004*. Ed. by M. Franklin. Vol. 3152. LNCS. Springer Berlin Heidelberg, 2004, pp. 56–72 (cit. on pp. 152–154, 170, 171, 179).

[CM11] S. Chatterjee and A. Menezes. "On cryptographic protocols employing asymmetric pairings — The role of Ψ revisited". In: *Discrete Applied Mathematics* 159.13 (2011), pp. 1311–1322 (cit. on p. 92).

[CP93] D. Chaum and T. P. Pedersen. "Wallet Databases with Observers". In: *Advances in Cryptology - CRYPTO '92*. Ed. by E. F. Brickell. Vol. 740. LNCS. Springer, 1993, pp. 89–105 (cit. on pp. 114, 157).

[CS97] J. Camenisch and M. Stadler. "Efficient group signature schemes for large groups". In: *Advances in Cryptology — CRYPTO '97*. Ed. by B. S. J. Kaliski. Vol. 1294. LNCS. Springer Berlin Heidelberg, 1997, pp. 410–424 (cit. on p. 101).

[Dam10] I. Damgård. *On Σ-protocols*. CPT 2010, v.2. 2010. URL: http://www.cs.au.dk/~ivan/Sigma.pdf (cit. on pp. 97, 100).

[Dam91] I. Damgård. "Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks". In: *Advances in Cryptology - CRYPTO '91*. Ed. by J. Feigenbaum. Vol. 576. LNCS. Springer, 1991, pp. 445–456 (cit. on pp. 173, 174).

[Den06] A. W. Dent. *The Hardness of the DHK Problem in the Generic Group Model*. Cryptology ePrint Archive, Report 2006/156. https://eprint.iacr.org/2006/156. 2006 (cit. on p. 175).

[DFH12] I. Damgård, S. Faust, and C. Hazay. "Secure Two-Party Computation with Low Communication". English. In: *Theory of Cryptography*. Ed. by R. Cramer. Vol. 7194. LNCS. Springer Berlin Heidelberg, 2012, pp. 54–74 (cit. on p. 174).

[DG09] M. Di Raimondo and R. Gennaro. "New Approaches for Deniable Authentication". English. In: *J. Cryptology* 22.4 (2009), pp. 572–615 (cit. on p. 174).

[DGK06] M. Di Raimondo, R. Gennaro, and H. Krawczyk. "Deniable Authentication and Key Exchange". In: *Proceedings of the 13th ACM Conference on Computer and Communications Security - CCS '06*. Alexandria, Virginia, USA: ACM, 2006, pp. 400–409 (cit. on p. 174).

[DH76] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654 (cit. on p. 90).

[DSD07] A. J. Devegili, M. Scott, and R. Dahab. "Implementing Cryptographic Pairings over Barreto-Naehrig Curves". In: *Pairing-Based Cryptography – Pairing 2007*. Ed. by T. Takagi, T. Okamoto, E. Okamoto, and T. Okamoto. Vol. 4575. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 197–207 (cit. on p. 93).

[DY05] Y. Dodis and A. Yampolskiy. "A Verifiable Random Function with Short Proofs and Keys". In: *Public Key Cryptography - PKC 2005*. Ed. by S. Vaudenay. Vol. 3386. LNCS. Springer Berlin Heidelberg, 2005, pp. 416–431 (cit. on p. 126).

[EMO09] K. Emura, A. Miyaji, and K. Omote. "A Certificate Revocable Anonymous Authentication Scheme with Designated Verifier". In: *Proceedings of the The Forth International Conference on Availability, Reliability and Security, ARES 2009*. IEEE Computer Society, 2009, pp. 769–773 (cit. on pp. 112, 121).

[EUDPD] *Data Protection Directive*. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. See also http://ec.europa.eu/justice/data-protection/. URL: http://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:31995L0046 (cit. on p. xvi).

[FHS15] G. Fuchsbauer, C. Hanser, and D. Slamanig. "Practical Round-Optimal Blind Signatures in the Standard Model". In: *CRYPTO (2)*. Vol. 9216. Lecture Notes in Computer Science. Springer, 2015, pp. 233–253 (cit. on p. 141).

[FS87] A. Fiat and A. Shamir. "How To Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *Advances in Cryptology – CRYPTO' 86*. Ed. by A. M. Odlyzko. Vol. 263. LNCS. Springer Berlin Heidelberg, 1987, pp. 186–194 (cit. on pp. 152, 169).

[FS90] U. Feige and A. Shamir. "Witness Indistinguishable and Witness Hiding Protocols". In: *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing - STOC '90*. Baltimore, Maryland, USA: ACM, 1990, pp. 416–426 (cit. on p. 99).

[Gal05] D. Galindo. "Boneh-Franklin Identity Based Encryption Revisited". In: *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005*. Ed. by L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung. Vol. 3580. LNCS. Springer, 2005, pp. 791–802 (cit. on p. 156).

[GK03] S. Goldwasser and Y. T. Kalai. "On the (In)security of the Fiat-Shamir Paradigm". In: *44th Symposium on Foundations of Computer Science - FOCS 2003*. IEEE Computer Society, 2003, pp. 102–113 (cit. on p. 169).

[GKZ14] R. Granger, T. Kleinjung, and J. Zumbrägel. "Breaking '128-bit Secure' Supersingular Binary Curves". In: *Advances in Cryptology – CRYPTO 2014*. Ed. by J. A. Garay and R. Gennaro. Vol. 8617. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, pp. 126–145 (cit. on p. 92).

[GMR88] S. Goldwasser, S. Micali, and R. L. Rivest. "A Digital Signature Scheme Secure Against Adaptive Chosen-message Attacks". In: *SIAM Journal on Computing* 17.2 (Apr. 1988), pp. 281–308 (cit. on pp. 102, 128).

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM Journal on Computing* 18.1 (1989), pp. 186–208 (cit. on p. 96).

[Gol00] O. Goldreich. *Foundations of Cryptography: Basic Tools*. New York, NY, USA: Cambridge University Press, 2000 (cit. on pp. 83, 96).

[Gol04] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. New York, NY, USA: Cambridge University Press, 2004 (cit. on p. 83).

[GPS08] S. D. Galbraith, K. G. Paterson, and N. P. Smart. "Pairings for cryptographers". In: *Discrete Applied Mathematics* 156.16 (2008), pp. 3113–3121 (cit. on pp. 93, 141).

[Gro07] J. Groth. "Fully Anonymous Group Signatures Without Random Oracles". In: *Advances in Cryptology – ASIACRYPT 2007*. Ed. by K. Kurosawa. Vol. 4833. LNCS. Springer Berlin Heidelberg, 2007, pp. 164–180 (cit. on p. 126).

[Gro10] J. Groth. "Short Pairing-Based Non-interactive Zero-Knowledge Arguments". English. In: *Advances in Cryptology - ASIACRYPT 2010*. Ed. by M. Abe. Vol. 6477. LNCS. Springer Berlin Heidelberg, 2010, pp. 321–340 (cit. on p. 174).

[HJV10] J.-H. Hoepman, B. Jacobs, and P. Vullers. "Privacy and Security Issues in e-Ticketing – Optimisation of Smart Card-based Attribute-proving". In: *Workshop on Foundations of Security and Privacy, FCS-PrivMod 2010*. Ed. by V. Cortier, M. Ryan, and V. Shmatikov. July 2010 (cit. on pp. 112, 114).

[HK14] L. Hanzlik and K. Kluczniak. "A Short Paper on How to Improve U-Prove Using Self-Blindable Certificates". In: *Financial Cryptography and Data Security: 18th International Conference, FC 2014*. Ed. by N. Christin and R. Safavi-Naini. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 273–282 (cit. on pp. xvi, 142–145, 147, 154, 194).

[HLR15] J.-H. Hoepman, W. Lueks, and S. Ringers. "On Linkability and Malleability in Self-blindable Credentials". In: *Information Security Theory and Practice: 9th IFIP WG 11.2 International Conference, WISTP 2015*. Ed. by N. R. Akram and S. Jajodia. Cham: Springer International Publishing, 2015, pp. 203–218 (cit. on pp. xvi, 111, 149, 194).

[HM13] J. Hajny and L. Malina. "Unlinkable Attribute-Based Credentials with Practical Revocation on Smart-Cards". English. In: *Smart Card Research and Advanced Applications*. Ed. by S. Mangard. Vol. 7771. LNCS. Springer Berlin Heidelberg, 2013, pp. 62–76 (cit. on p. 153).

[IBM12] IBM Research Zürich Security Team. *Specification of the Identity Mixer Cryptographic Library, version 2.3.0*. Tech. rep. IBM Research, Zürich, Feb. 2012. URL: https://tinyurl.com/idemix-spec (cit. on pp. 112, 123, 151, 153).

[JC97] W.-S. Juang and L. Chin-Laung. "A secure and practical electronic voting scheme for real world environments". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 80.1 (1997), pp. 64–71 (cit. on p. 125).

[JLO97] A. Juels, M. Luby, and R. Ostrovsky. "Security of blind digital signatures". In: *Advances in Cryptology — CRYPTO '97*. Ed. by B. S. Kaliski. Vol. 1294. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 150–164 (cit. on p. 140).

[Jou00] A. Joux. "A One Round Protocol for Tripartite Diffie-Hellman". In: *Proceedings of the 4th International Symposium on Algorithmic Number Theory*. ANTS-IV. London, UK, UK: Springer-Verlag, 2000, pp. 385–394. URL: http://dl.acm.org/citation.cfm?id=648185.749894 (cit. on p. 92).

[Kra05] H. Krawczyk. "HMQV: A High-Performance Secure Diffie-Hellman Protocol". English. In: *Advances in Cryptology – CRYPTO 2005*. Ed. by V. Shoup. Vol. 3621. LNCS. Springer Berlin Heidelberg, 2005, pp. 546–566 (cit. on p. 174).

[KT08] S. Kiyomoto and T. Tanaka. "Anonymous attribute authentication scheme using self-blindable certificates". In: *IEEE International Conference on Intelligence and Security Informatics, ISI 2008*. IEEE, 2008, pp. 215–217 (cit. on pp. 112, 120, 121).

[Len05] A. K. Lenstra. "Key Lengths". In: *Handbook of Information Security*. Ed. by H. Bidgoli. Wiley, 2005 (cit. on p. 91).

[Lin11] Y. Lindell. *Sigma Protocols and Zero-Knowledge*. Lecture notes from Winter School on Secure Computation and Efficiency. 2011. URL: http://u.cs.biu.ac.il/~lindell/winterschool2011/lecture%205.pdf (cit. on p. 100).

[Lue+15] W. Lueks, G. Alpár, J.-H. Hoepman, and P. Vullers. "Fast Revocation of Attribute-Based Credentials for Both Users and Verifiers". In: *ICT Systems Security and Privacy Protection: 30th IFIP TC 11 International Conference, SEC 2015*. Ed. by H. Federrath and D. Gollmann. Cham: Springer International Publishing, 2015, pp. 463–478 (cit. on p. 105).

[LV01] A. K. Lenstra and E. R. Verheul. "Selecting Cryptographic Key Sizes". In: *J. Cryptology* 14.4 (2001), pp. 255–293 (cit. on pp. 91, 171).

[Lys+00] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. "Pseudonym Systems". In: *Selected Areas in Cryptography: 6th Annual International Workshop, SAC'99*. Ed. by H. Heys and C. Adams. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 184–199 (cit. on pp. 152, 154).

[Lys99] A. Lysyanskaya. "Pseudonym Systems". MSc thesis. Massachusetts Institude of Technology, 1999. URL: https://groups.csail.mit.edu/cis/theses/anna-sm.pdf (cit. on pp. 152, 154).

[MNT01] A. Miyaji, M. Nakabayashi, and S. Takano. "New explicit conditions of elliptic curve traces for FR-reduction". In: *IEICE transactions on fundamentals of electronics, communications and computer sciences* 84.5 (2001), pp. 1234–1243 (cit. on p. 93).

[MPR11] H. K. Maji, M. Prabhakaran, and M. Rosulek. "Attribute-Based Signatures". In: *Topics in Cryptology – CT-RSA 2011*. Ed. by A. Kiayias. Vol. 6558. LNCS. Springer Berlin Heidelberg, 2011, pp. 376–392 (cit. on p. 126).

[Nao03] M. Naor. "On Cryptographic Assumptions and Challenges". In: *Advances in Cryptology - CRYPTO 2003*. Ed. by D. Boneh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 96–109 (cit. on pp. 153, 154, 174).

[Oka06] T. Okamoto. "Efficient Blind and Partially Blind Signatures Without Random Oracles". In: *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006*. Ed. by S. Halevi and T. Rabin. Vol. 3876. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 80–99 (cit. on p. 141).

[Pai99] P. Paillier. "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes". In: *Advances in Cryptology — EUROCRYPT '99*. Ed. by J. Stern. Vol. 1592. LNCS. Springer Berlin Heidelberg, 1999, pp. 223–238 (cit. on pp. 129, 130).

[Pap94] C. M. Papadimitriou. *Computational complexity*. Reading, Massachusetts: Addison-Wesley, 1994 (cit. on pp. 83, 86).

[Poi98] D. Pointcheval. "Strengthened security for blind signatures". In: *Advances in Cryptology — EUROCRYPT'98*. Ed. by K. Nyberg. Vol. 1403. LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 391–405 (cit. on p. 141).

[PZ13] C. Paquin and G. Zaverucha. "U-Prove Cryptographic Specification V1.1 (Revision 3)". Released under the Open Specification Promise. Dec. 2013. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=166969 (cit. on pp. 126, 151, 153).

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-key Cryptosystems". In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126 (cit. on p. 87).

[Sch90] C. Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *Advances in Cryptology — EUROCRYPT '89*. Ed. by J.-J. Quisquater and J. Vandewalle. Vol. 434. LNCS. Springer Berlin Heidelberg, 1990, pp. 688–689 (cit. on p. 98).

[Sho97] V. Shoup. "Lower Bounds for Discrete Logarithms and Related Problems". In: *Advances in Cryptology — EUROCRYPT '97*. Ed. by W. Fumy. Vol. 1233. LNCS. Springer Berlin Heidelberg, 1997, pp. 256–266 (cit. on pp. 154, 175).

[SV07a] N. P. Smart and F. Vercauteren. "On computable isomorphisms in efficient asymmetric pairing-based systems". In: *Discrete Applied Mathematics* 155.4 (2007), pp. 538–547. URL: http://dx.doi.org/10.1016/j.dam.2006.07.004 (cit. on p. 156).

[SV07b] N. Smart and F. Vercauteren. "On computable isomorphisms in efficient asymmetric pairing-based systems". In: *Discrete Applied Mathematics* 155.4 (2007), pp. 538–547 (cit. on pp. 103, 157).

[Tur37] A. M. Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem". In: *Proc. London Math. Soc. Ser.* 2 (42 1937), pp. 230–265 (cit. on pp. 83, 89).

[VA13] P. Vullers and G. Alpár. "Efficient Selective Disclosure on Smart Cards Using Idemix". In: *Policies and Research in Identity Management*. Ed. by S. Fischer-Hübner, E. de Leeuw, and C. Mitchell. Vol. 396. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, 2013, pp. 53–67 (cit. on p. 153).

[Ver01] E. R. Verheul. "Self-Blindable Credential Certificates from the Weil Pairing". In: *Advances in Cryptology - ASIACRYPT*. Ed. by C. Boyd. Vol. 2248. LNCS. Springer, 2001, pp. 533–551 (cit. on pp. 112–114, 144, 147, 148, 151, 152, 158).

[VRH16] E. Verheul, S. Ringers, and J.-H. Hoepman. "The self-blindable U-Prove scheme from FC'14 is forgeable". In: *Financial Cryptography and Data Security – FC'16* (2016). In print. URL: https://eprint.iacr.org/2015/725 (cit. on pp. xvi, 143, 194).

[Wac+11] C. Wachsmann, L. Chen, K. Dietrich, H. Löhr, A.-R. Sadeghi, and J. Winter. "Lightweight Anonymous Authentication with TLS and DAA for Embedded Mobile Devices". In: *Information Security: 13th International Conference, ISC 2010*. Ed. by M. Burmester, G. Tsudik, S. Magliveras, and I. Ilić. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 84–98. URL: https://eprint.iacr.org/2011/101.pdf (cit. on pp. 153, 154).

[Wat05] B. Waters. "Efficient Identity-Based Encryption Without Random Oracles". In: *EUROCRYPT*. Vol. 3494. Lecture Notes in Computer Science. Springer, 2005, pp. 114–127 (cit. on p. 141).

[Wei12] E. Weitenberg. "Providing unlinkability of transactions with a single token in U-Prove". MSc thesis. University of Groningen, July 2012. URL: http://scripties.fwn.eldoc.ub.rug.nl/scripties/Wiskunde/Masters/2012/Weitenberg.E.R.A./ (cit. on p. 141).

[WS07] J. Wu and D. R. Stinson. "An Efficient Identification Protocol and the Knowledge-of-Exponent Assumption". In: *IACR Cryptology ePrint Archive* 2007 (2007), p. 479. URL: https://eprint.iacr.org/2007/479 (cit. on p. 174).

[WS08] J. Wu and D. Stinson. *On The Security of The ElGamal Encryption Scheme and Damgård's Variant*. Cryptology ePrint Archive, Report 2008/200. 2008. URL: https://eprint.iacr.org/2008/200 (cit. on p. 174).

[WY05] V. K. Wei and T. H. Yuen. "More short signatures without random oracles". In: *IACR Cryptology ePrint Archive* 2005 (2005), p. 463. URL: https://eprint.iacr.org/2005/463 (cit. on pp. 153–155).

[YZ10] A. C. Yao and Y. Zhao. "Deniable Internet Key Exchange". English. In: *Applied Cryptography and Network Security*. Ed. by J. Zhou and M. Yung. Vol. 6123. LNCS. Springer Berlin Heidelberg, 2010, pp. 329–348 (cit. on p. 174).

[ZSS03] F. Zhang, R. Safavi-Naini, and W. Susilo. "Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings". In: *Progress in Cryptology - INDOCRYPT 2003*. Ed. by T. Johansson and S. Maitra. Vol. 2904. LNCS. Corrected version at https://www.uow.edu.au/~wsusilo/VESPBS.pdf. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 191–204 (cit. on p. 141).

List of notations

Quantization using Jet Space Geometry

π_{BV}	π adjoined with its parity-reversed dual, p. 36
$U_\hbar(\mathfrak{g})$	$U(\mathfrak{g})[\![\hbar]\!]$ with $[\,,\,]$ rescaled to $\hbar[\,,\,]$, p. 65
Δ	(BV) Laplacian, p. 28
$[\![\,,\,]\!]$	(variational) Schouten bracket, p. 16
$\widehat{P}$	adjoint of module P, p. 9
λ^Σ	antisymmetrization of cocycle λ, p. 58
π_∞	bundle projection $\pi_\infty : J^\infty(\pi) \to M$ induced by π, p. 4
$\mathrm{d}_\mathcal{C}$	Cartan differential, p. 6
$\widehat{\varkappa}(\pi)$	covectors on $J^\infty(\pi)$, p. 7
$\lvert\xi\rvert$	degree of multivector or field ξ, p. 13
$\bar{\Lambda}^k(\pi)$	differential forms, horizontal, of degree k on $J^\infty(\pi)$, p. 6
$\mathcal{C}^k\Lambda(\pi)$	differential forms, vertical, of degree k on $J^\infty(\pi)$, p. 6
$U(\mathfrak{g})$	enveloping algebra of $\mathfrak{g}$, p. 63
δ_q	Euler operator w.r.t. fiber coordinates q^α, p. 8
δ	Euler operator, p. 7
$\boldsymbol{\pi}$	even part of the BV superbundle, p. 39
$\partial_\varphi^{(q)}$	evolutionary vector field w.r.t fiber coordinates q^α with generator φ, p. 6
$\varkappa(\pi)$	evolutionary vector fields on $J^\infty(\pi)$, p. 6
$\mathfrak{M}(\pi)$	formal products of integral functionals, p. 34
X_H	Hamiltonian vector field associated with function H, p. 29
d^H	Hochschild differential, p. 56

$\bar{H}^i(\pi)$	horizontal i-forms modulo the horizontal differential $\bar{\mathrm{d}}$, p. 7
$\bar{\mathrm{d}}$	horizontal differential, p. 6
$\overline{J^\infty_\pi}(\xi)$	horizontal jet bundle of ξ over π, p. 8
$h([q])$	indicates dependence of h on jet coordinates q^α_σ, p. 33
$J^\infty(\pi)$	infinite jet bundle induced by the bundle π, p. 4
$j^\infty_x(s)$	jet of section $s \in \Gamma(\pi)$ evaluated at $x \in M$, p. 4
$j^\infty(s)$	jet of section $s \in \Gamma(\pi)$, p. 4
$\frac{\overrightarrow{\partial}}{\partial b_{\alpha,\sigma}}$	left derivative w.r.t. odd fiber coordinate $b_{\alpha,\sigma}$, p. 16
Π	parity reversion operator of fiber bundles, p. 14
$\hbar$	Planck constant, or: deformation parameter, p. 45
$\frac{\overleftarrow{\partial}}{\partial b_{\alpha,\sigma}}$	right derivative w.r.t. odd fiber coordinate $b_{\alpha,\sigma}$, p. 16
$\mathcal{F}(\pi)$	ring of smooth functions on $J^\infty(\pi)$, p. 4
$T_{\text{multi}}(M)$	shifted space of multivector fields, p. 56
$\star$	star product, p. 48
$S(\mathfrak{g})$	symmetric algebra of $\mathfrak{g}$, p. 64
D_i	total derivative w.r.t. base coordinate x^i, p. 5
D_σ	total derivatives w.r.t. multi-index σ, p. 5
$\mathcal{C}\mathrm{Diff}_k(P, P')$	total differential operator from P to P' taking k arguments, p. 10
$\frac{\delta}{\delta q^\alpha}$	variational derivative w.r.t. fiber coordinate q^α, p. 7

Identity Management using Credential Schemes

$\mathrm{negl}(\ell)$	a function negligible in ℓ, p. 86
$\mathcal{A}^O$	algorithm $\mathcal{A}$ has oracle access to oracle O, p. 86
$\mathcal{A}(x) \to y$	algorithm $\mathcal{A}$ outputs y on input x, p. 86
$\mathcal{B}\blacksquare\!\!\to\mathcal{A}$	algorithm $\mathcal{B}$ has black-box access to algorithm $\mathcal{A}$, p. 95
$\mathcal{A}(x) \leftrightarrow \mathcal{B}(y)$	algorithms $\mathcal{A}$ and $\mathcal{B}$ interacting on inputs x and y respectively, p. 86
k_i	attribute i of an attribute-based credential, p. 106
$k_{i,j}$	attribute i of the j-th attribute-based credential, p. 106
$e(\cdot, \cdot)$	bilinear pairing, p. 91
$\mathrm{PK}\{\dots\}$	Camenisch-Stadler notation for zero-knowledge proofs, p. 101
$X \overset{c}{\approx} Y$	computational indistinguishability of X and Y, p. 94
Sig_{SK}	function mapping secret keys and blinding factors to signatures, p. 115
PubKey	function mapping secrets keys and blinding factors to public keys, p. 114
$\mathbb{G}$	group family, p. 86
$\mathcal{D}$	index set of disclosed attributes $\subset \{1, \dots, n\}$, p. 106
$\mathcal{C}$	index set of secret attributes $\subset \{1, \dots, n\}$, p. 106

$\mathbb{Z}_n$	integers modulo n, i.e., $\mathbb{Z}/n\mathbb{Z}$, p. 88
Issue	issuing protocol for credential scheme, p. 105
KeyGen	key generation algorithm (for signature, credential or encryption schemes), p. 101
$\mathcal{M}$	message space of a signature scheme, p. 101
$\mathbb{Z}_n^*$	multiplicative subgroup of integers modulo n, p. 88
$\mathcal{P}$	prover or user in zero-knowledge proofs or credential schemes, p. 95
ℓ	security parameter, p. 86
$\mathcal{B}$	set of blinding factors, p. 113
$\mathcal{C}$	set of credentials, p. 113
$\mathcal{P}$	set of private keys, p. 113
$\mathcal{K}$	set of public keys, p. 113
$\mathcal{S}$	set of signatures, p. 113
ShowCredential	showing protocol for credential scheme, p. 105
Sign_{SK}	signing algorithm of a signature scheme, p. 101
$\mathcal{S}$	simulator for zero-knowledge proofs, p. 96
Verify_{PK}	verification algorithm of a signature scheme, p. 101
$\mathcal{V}$	verifier in zero-knowledge proofs or credential schemes, p. 95
χ	witness extractor for zero-knowledge proofs of knowledge, p. 96

Index

Quantization using Jet Space Geometry

Identity Management using Credential Schemes